entropy

MDPI

Article

Making the Coupled Gaussian Process Dynamical Model Modular and Scalable with Variational Approximations †

Dmytro Velychko *, Benjamin Knopp and Dominik Endres *

Department of Psychology, University of Marburg, Gutenbergstr. 18, 35032 Marburg, Germany; benjamin.knopp@uni-marburg.de

* Correspondence: dmytro.velychko@uni-marburg.de (D.V.); dominik.endres@uni-marburg.de (D.E.)

† This paper is an extended version of our paper published in the 26th International Conference on Artificial Neural Networks (ICANN 2017), Alghero, Italy, 11–14 September, 2017.

Received: 27 July 2018; Accepted: 20 September 2018; Published: 21 September 2018

Abstract: We describe a sparse, variational posterior approximation to the Coupled Gaussian Process Dynamical Model (CGPDM), which is a latent space coupled dynamical model in discrete time. The purpose of the approximation is threefold: first, to reduce training time of the model; second, to enable modular re-use of learned dynamics; and, third, to store these learned dynamics compactly. Our target applications here are human movement primitive (MP) models, where an MP is a reusable spatiotemporal component, or "module" of a human full-body movement. Besides re-usability of learned MPs, compactness is crucial, to allow for the storage of a large library of movements. We first derive the variational approximation, illustrate it on toy data, test its predictions against a range of other MP models and finally compare movements produced by the model against human perceptual expectations. We show that the variational CGPDM outperforms several other MP models on movement trajectory prediction. Furthermore, human observers find its movements nearly indistinguishable from replays of natural movement recordings for a very compact parameterization of the approximation.

Keywords: Gaussian processes; variational methods; movement primitives; modularity

1. Introduction

Two formidable problems that the human brain has to solve are planning and execution of movements of its body. As a means to simplify these problems while keeping a sufficient degree of control flexibility for a wide range of tasks, modular movement primitives (MP) have been suggested (see [1,2] for reviews). There is no universally accepted definition of the term "movement primitive". For the purposes of this paper, an MP is a spatiotemporal component of a human (full-body) movement that may be produced by mapping a latent state onto observable variables, such as joint angles. The latent state can be generated by dynamical systems [3] or source functions [4,5]. "Modular" usually refers to the existence of an operation which allows for the spatial, temporal or spatiotemporal combination of (simple) primitives into (complex) movements.

Two prominent examples, where this operation is the linear combination of stereotypical time-courses or muscle-coactivations, are called temporal MP-models [6–9] or spatial MP-models [10,11]. While these models are inherently modular, the assumption of stereotyped MPs makes it difficult for a control system built out of these primitives to respond to perturbations. A type of MP which can be controlled on-line more easily is the dynamical MP (DMP) [3], which has been developed for robotics applications. In this approach, each primitive is encoded by a canonical second order differential equation with guaranteeable stability properties and learnable parameters. A DMP can generate both discrete (e.g., reaching) and rhythmic (e.g., walking) movements and drives the trajectory of one degree of freedom, e.g., a joint angle. Modularity arises because of the latter property, which might be viewed as an "extreme" form of the modularization that we investigate here, where one movement module might affect several degrees of freedom. Similarly, recent extensions of the DMP framework allow for the reuse of a DMP across end-effectors via kinematical mappings [12] or across tasks [13].

We describe a model that learns MPs composed of coupled dynamical systems and associated kinematics mappings, where both components are learned, thus lifting the DMP's restriction of canonical dynamics. We build on the Coupled Gaussian Process Dynamical Model (CGPDM) by [14], which combines the advantages of modularity and flexibility in the dynamics, at least theoretically. In a CGPDM, the temporal evolution functions for the latent dynamical systems are drawn out of a Gaussian process (GP) prior [15]. These dynamical systems are then coupled probabilistically, and the result is mapped onto observations by functions drawn from another GP. One drawback of the CGPDM is its fully non-parametric nature, which results in cubic scaling (with the dataset size) of learning complexity and quadratic scaling of MP storage size, i.e., the CGPDM can not be learned from large data sets, and its effective parameterization is not compact. We improve both scalability and compactness with deterministic, sparse variational approximations [16]. In this sparse variational CGPDM, each MP is parameterized by a small set of inducing points (IPs) and associated inducing values (IVs), leading to a compact representation with linear scaling of the training complexity in the number of data points, and constant storage requirements. This compactness is important for real-world applicability of the model, since there might be more primitives than muscles (or actuators) across tasks, as pointed out by Bizzi and Cheung [17]: the motor "code" might be sparse and overcomplete, similar to the sparse codes in early vision [18]. Table 1 provides an overview of the key MP models which we compare in this paper.

Table 1. Overview of movement primitive models compared in this paper. (v)CGPDM, (variational) coupled Gaussian process dynamical model; (v)GPDM, (variational) Gaussian process dynamical model; TMP, temporal movement primitives; DMP, dynamical movement primitives. Modular, learns reusable MPs. Scalable, below cubic learning complexity with respect to the data set size; Compact, size of the effective parameterization does not grow with the data set size; Canonical dynamics, dynamics model specified before learning; Learned dynamics, dynamics model is a free-form function.

	Modular	**Scalable**	**Compact**	**Canonical Dynamics**	**Learned Dynamics**
vCGPDM	✓	✓	✓	x	✓
CGPDM	x	x	x	x	✓
vGPDM	x	✓	✓	x	✓
GPDM	x	x	x	x	✓
TMP	✓	✓	✓	x	x
DMP	✓	✓	✓	✓	x

Our target application here is human movement modeling, but the vCGPDM could be easily applied to other systems where modularized control is beneficial, e.g., humanoid robotics [9].

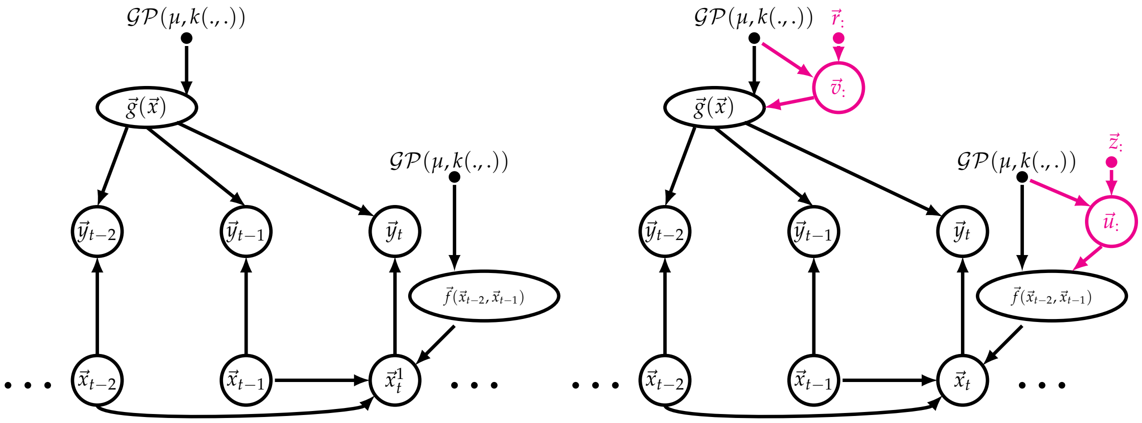
We briefly review related work in Section 2 and introduce the vCGPDM in Section 3. The derivation of the variational approximation is outlined in Section 4. In Section 5, we first illustrate the vCGPDM on artificial data. Second, we benchmark the vCGPDM against other MP models. Third, we perform an experiment to quantify the degree of of human-tolerable sparseness in a psychophysics experiment. Fourth, we demonstrate modular movement composition with the vCGPDM. In Section 6, we propose future research directions based on our work.

This paper is a substantially extended version of our earlier conference publication [19].

2. Related Work

The Gaussian process (GP) is a machine learning staple for classification and regression tasks [15]. A GP is a prior on functions $\mathbb{R}^Q \to \mathbb{R}$ from a Q-dimensional input space to one-dimensional output. By drawing D times from the GP, functions from $\mathbb{R}^Q \to \mathbb{R}^D$ can be realized. Its advantages include theoretical elegance, tractability and closed-form solutions for posterior densities. Its main disadvantage is cubic runtime scaling with the number of data points. Several solutions have been proposed for this problem. Many of these involve a sparse representation of the posterior process via a small set of IPs, which may [20] or may not be a subset of the data points [21]. If the input space is unobserved, one obtains a GP latent variable model (GPLVM), for which sparse approximations have also been devised [22]. One problem with sparse GP approximations is their tendency to overfit [22], leading to incorrect variance predictions [23]. In that paper, it is also demonstrated that the problem can be alleviated by a variational approximation, which prompted us to develop a similar approach for the CGPDM: as in [24], we extend the sparse GPLVM in time, but we use an autoregressive dynamical system.

If the temporal evolution function of this dynamical system is also drawn from a GP, the resulting model is called Gaussian Process Dynamical Model (GPDM), which can be learned by maximum-a-posteriori approximation if the observed dimension D is greater than the latent dimension Q [25]. Figure 1 (left) shows a graphical model representation of the GPDM, and introduces the related notation which we use throughout the paper. Slices ":" indicate collections of variables along one integer index. Multiple slices refer to collections along multiple indices, e.g., $\vec{x}^{:}_{:}$ are the latent variables of all parts and time-steps. The GPDM can model the variability of human movements and has been used for computer animation with style control [26–28]. It has also been used with an additional switching prior on the dynamics for motion tracking and recognition [29] and deep variants have been devised [30]. However, with the exception of the coupled GPDM [14,19], all these approaches have a "monolithic" latent space and thus lack the modularity of MPs. One reason for this might be the fact that, for the maximum-a-posteriori approximation to work, the latent space has to be lower-dimensional than the observed space, $Q \ll D$. If, as explained above, we want a modular, possibly overcomplete (i.e., the effective $Q > D$) set of MPs, we need learning approaches that are robust to overfitting. The works of Frigola et al. [31] indicate that such approaches may be obtained with variational approximations. In the following, we therefore introduce a variational approximation to CGPDM learning and inference based on an approach similar to Frigola et al. [32], but, as in [30], we aim to obviate the need for sampling altogether to allow for fast, repeatable trajectory generation. While deriving a variational approximation is not trivial, we expect it to avoid overfitting and yield a good bound on the marginal likelihood [33]. Figure 1 (right) shows the graphical model of the GPDM augmented by IPs and IVs. This augmentation yields a tractable variational approximation to the GPDM's posterior [30].



Notation and Abbreviations (v)GPDM

Gaussian process dynamical model	GPDM	variational approximation to posterior of GPDM	vGPDM
discrete time index	$t = 1, \ldots, T$	mean and kernel function	$\mu,\ k(.,.)$
latent space dimensionality	Q	latent states	$\vec{x}_t \in \mathbb{R}^Q$
observed space dimensionality	D	observable variables	$\vec{y}_t \in \mathbb{R}^D$
Gaussian process prior	$\mathcal{GP}(\mu, k(.,.))$	Inducing points/values	IP/IV
latent-to-observed function	$\vec{g}(\vec{x}_t)$	dynamics function	$\vec{f}(\vec{x}_{t-2}, \vec{x}_{t-1})$
IP of latent-to-observed function	$\vec{r}_: = (\vec{r}_1, \vec{r}_2, \ldots)$	IP of dynamics function	$\vec{z}_: = (\vec{z}_1, \vec{z}_2, \ldots)$
IV of latent-to-observed function	$\vec{v}_: = (\vec{v}_1, \vec{v}_2, \ldots)$	IV of dynamics function	$\vec{u}_: = (\vec{u}_1, \vec{u}_2, \ldots)$

Figure 1. Modular building blocks of the vCGPDM. (**Left**) The Gaussian process dynamical model (GPDM). A latent, second order dynamics model generates a time-series of vector-valued random variables $\vec{x}_t$ which are drawn from a multivariate Gaussian distribution with mean function $\vec{f}(\vec{x}_{t-2}, \vec{x}_{t-1})$. The components of this mean function are drawn from a Gaussian Process $\mathcal{GP}(\mu, k(.,.))$. Each observable $\vec{y}_t$ is drawn from multivariate Gaussian distribution with mean function $\vec{g}(\vec{x}_t)$, which have a Gaussian process prior, too. (**Right**) The GPDM augmented with inducing points and values for a sparse representation of the posterior process [23]. This enables faster variational Bayesian learning and inference, because the augmented GPs are effectively parameterized by these points (here, $\vec{r}_:$, $\vec{z}_:$) and corresponding values (here, $\vec{v}_:$,$\vec{u}_:$) rather than by the full dataset. They may be thought of as prototypical examples of the corresponding functions, e.g., $\vec{v}_k = \vec{g}(\vec{r}_k)$. Slice notation ":" indicates collections of variables. For details, see text.

3. The Model

The basic building blocks, or "parts", of the CGPDM are a number of GPDMs run in parallel. In the context of human movement modeling, e.g., they may be thought of as body parts. A part evolves in discrete time $t = 0, \ldots, T$ and is endowed with a Q-dimensional latent space, a D-dimensional observed space and second-order autoregressive dynamics described by a function $\vec{f}(\vec{x}_{t-2}, \vec{x}_{t-1})$. The component functions $\left(\vec{f}(\vec{x}_{t-2}, \vec{x}_{t-1})\right)_q$ have a Gaussian process prior $\mathcal{GP}(\mu, k(.,.))$ with mean function μ and kernel $k(.,.)$ Second-order dynamics seem to be a good choice for our target application of human movement modeling [34], but the order can be easily altered simply by concatenating previous states into one larger vector. Let $\vec{x}_t \in \mathbb{R}^Q$ the state of latent space of the part at time t (see Figure 1, left). This latent state produces observations $\vec{y}_t \in \mathbb{R}^D$ via the function $\vec{g}(\vec{x}_t)$ as well as isotropic Gaussian noise with variance β. The components $(\vec{g}(\vec{x}_t))_d$ of this function are drawn from a Gaussian process prior, too. GPDMs can be learned from data via a combination of exact marginalization and maximum-a-posteriori learning of the latent dynamics [25].

While the GPDM is a very expressive model, it suffers from poor runtime and memory scaling with the data set size due to its non-parametric nature, which it inherits from the involved GPs. We remedied this problem by an approach pioneered in [23]: augmenting the GPs with inducing points (IPs, here: $\vec{r}_:$, $\vec{z}_:$) and associated inducing values (IVs, $\vec{v}$, $\vec{u}$) (see Figure 1, right). These IP/IV pairs might be thought of as prototypical examples of the mappings represented by the corresponding functions, e.g., $\vec{v}^i_k = \vec{g}^i(\vec{r}^i_k)$. Note that the IPs are not drawn from a prior, whereas the IVs are. Hence, the latter are model parameters, whereas the former are not: IPs are merely parameters of the approximation, or "variational parameters". This augmentation allows for the derivation of a closed-form evidence lower bound (ELBO) on the marginal likelihood of the model.

In a CGPDM, the latent spaces of the parts are coupled to each other. We index parts by superscripts $i = 1, \ldots, M$. The index notation in such models can be confusing for first-time readers, we provide a notation and index overview in the tables below Figure 2. In our target application, the coupling may reflect the influences which parts of an articulated body have on each other during the execution of a movement. The coupling is implemented by having the parts make Gaussian-distributed predictions $\vec{x}^{i,j}_t$ about each other's latent states with means generated by $M \times M$ many mean coupling functions $\vec{f}^{i,j}(\vec{x}^i_{t-2}, \vec{x}^i_{t-1})$ and coupling variances $\alpha^{i,j}$. i indexes the origin part of a coupling, and j its target. Thus, $\vec{f}^{i,i}$ refers to the dynamics function for part i. The components of $\vec{f}^{i,j}(\vec{x}^i_{t-2}, \vec{x}^i_{t-1})$ are drawn from GPs. As described in [14], these predictions are combined with a product-of-experts construction (PoE [35]), including the predictions which a part makes about its own future. A product-of-experts construction forces the experts to agree on one prediction (Equation (1), left), which amounts to multiplying the individual predictions (Equation (1), right) and renormalizing (Equation (1), middle):

$$p(\vec{x}^j_t | \vec{f}^{:,j}(\vec{x}^i_{t-2}, \vec{x}^j_{t-1}), \alpha^{:,j}) = \frac{\exp\left[-\frac{1}{2\alpha^j} \left(\vec{x}^j_t - \alpha^j \sum_i \frac{\vec{f}^{i,j}(\vec{x}^i_{t-2}, \vec{x}^j_{t-1})}{\alpha^{i,j}} \right)^2 \right]}{(2\pi\alpha^j)^{\frac{Q^j}{2}}} \propto \prod_i \mathcal{N}\left(\vec{x}^j_t | \vec{f}^{i,j}(\vec{x}^i_{t-2}, \vec{x}^i_{t-1}), \alpha^{i,j} \right) \tag{1}$$

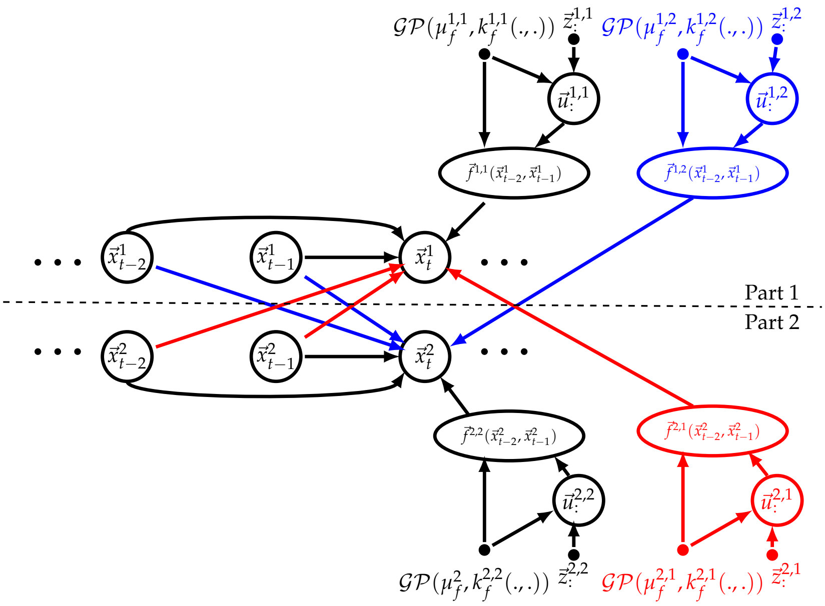
where $\alpha^j = \left(\sum_i (\alpha^{i,j})^{-1} \right)^{-1}$.

To understand the function of the $\alpha^{i,j}$, consider the form of Equation (1): the smaller a given variance, the more important the prediction of the generating part. We optimize the $\alpha^{i,j}$ during learning, letting the model discover which couplings are important for predicting the data. In other words, whenever an $\alpha^{i,j}$ is small compared to $\alpha^{i' \neq i,j}$, then part i is able to make a prediction about part j with (relatively) high certainty. Furthermore, the $\alpha^{i,j}$ can be modulated after learning to generate new movements, as shown below.

In the following, we denote all relevant timesteps before time t with subscript $-t$, e.g., $\vec{x}^j_{-t} = (\vec{x}^j_{t-2}, \vec{x}^j_{t-1})$ for a second-order dynamics model. We showed in [14] that the individual predictions of part i about part j, $\vec{x}^{i,j}_t$ can be exactly marginalized, leading to a GPDM-like model for each part with a dynamics kernel given by the $\alpha_{i,j}$-weighted mean of the individual coupling kernels:

$$k^j_f\left(\vec{x}^1_{-t}, \vec{x}^{1\prime}_{-t}, \ldots, \vec{x}^M_{-t}, \vec{x}^{M\prime}_{-t} \right) = \alpha^{j^2} \sum_{i=1}^{M} \frac{k^{i,j}_f(\vec{x}^i_{-t}, \vec{x}^{i\prime}_{-t}, \vec{x}^j_{-t}, \vec{x}^{j\prime}_{-t})}{\alpha^{i,j^2}} \tag{2}$$

However, doing so results in a model which lacks modularity: after learning, it is difficult to separate the parts from each other, and recombine them for the production of new movements that were not in the training data. We facilitate this modular recombination by restating CGPDM learning such that we can keep an explicit, sparse representation of the coupling functions. Another reason for a sparse representation is that the CGPDM exhibits cubic run time scaling with the data points, which it inherits from the composing GPDMs. To remedy these problems, we follow the treatment in [16,30]: we augment the model with IPs $\vec{r}^i_:$ and associated IVs $\vec{v}^i_:$ such that $g^i(\vec{r}^i_k) = \vec{v}^i_k$ for the latent-to-observed mappings $g^i()$. Then, we condition the probability density of the function values of $g^i()$ on these IPs/IVs, which we assume to be a sufficient statistic. Likewise, we reduce the computational effort for learning the dynamics and coupling mappings by inducing them through $\vec{z}^{i,j}_:$ and $\vec{u}^{i,j}_:$ (also known as "dynamics IPs/IVs"). See Figure 2 for a graphical representation of the augmented model.


$\mathcal{GP}(\mu_f^{1,1}, k_f^{1,1}(.,.))$
$\vec{z}_:^{1,1}$
$\mathcal{GP}(\mu_f^{1,2}, k_f^{1,2}(.,.))$
$\vec{z}_:^{1,2}$
$\vec{u}_:^{1,1}$
$\vec{u}_:^{1,2}$
$\vec{f}^{1,1}(\vec{x}_{t-2}^1, \vec{x}_{t-1}^1)$
$\vec{f}^{1,2}(\vec{x}_{t-2}^1, \vec{x}_{t-1}^1)$
$\vec{x}_{t-2}^1$
$\vec{x}_{t-1}^1$
$\vec{x}_t^1$
Part 1
Part 2
$\vec{x}_{t-2}^2$
$\vec{x}_{t-1}^2$
$\vec{x}_t^2$
$\vec{f}^{2,2}(\vec{x}_{t-2}^2, \vec{x}_{t-1}^2)$
$\vec{f}^{2,1}(\vec{x}_{t-2}^2, \vec{x}_{t-1}^2)$
$\vec{u}_:^{2,2}$
$\vec{u}_:^{2,1}$
$\mathcal{GP}(\mu_f^2, k_f^{2,2}(.,.))$
$\vec{z}_:^{2,2}$
$\mathcal{GP}(\mu_f^{2,1}, k_f^{2,1}(.,.))$
$\vec{z}_:^{2,1}$

Notation and Abbreviations (v)CGPDM

Coupled Gaussian process dynamical model	CGPDM	variational approximation to posterior of CGPDM	vCGPDM
latent states, part i at time t	$\vec{x}_t^i \in \mathbb{R}^Q$	Inducing points/values	IP/IV
latent states, part i before t	$\vec{x}_{-t}^i$	latent predictions of part j about part i at time t	$\vec{x}_t^{j,i}$
observable variables, part i at time t	$\vec{y}_t^i \in \mathbb{R}^D$	Gaussian process prior	$\mathcal{GP}(\mu, k(.,.))$
dynamics function, part i	$\vec{f}^{i,i}(\vec{x}_{t-2}^i, \vec{x}_{t-1}^i)$	latent-to-observed function, part i	$\vec{g}^i(\vec{x}_t^i)$
coupling function, part j-to-i	$\vec{f}^{j,i}(\vec{x}_{t-2}^j, \vec{x}_{t-1}^j)$	coupling variance, part j-to-i	$\alpha^{j,i}$
mean and kernel of $\mathcal{GP}$ prior on dynamics function, part i	$\mu_f^i,\ k_f^{i,i}(.,.)$	mean and kernel of $\mathcal{GP}$ prior on latent-to-observed function, part i	$\mu_g^i,\ k_g^{i,i}(.,.)$
IP of latent-to-observed function, part i	$\vec{r}_:^i = (\vec{r}_1^i, \vec{r}_2^i, \ldots)$	IP of dynamics function, part j-to-i	$\vec{z}_:^{j,i} = (\vec{z}_1^{j,i}, \vec{z}_2^{j,i}, \ldots)$
IV of latent-to-observed function, part i	$\vec{v}_:^i = (\vec{r}_1^i, \vec{r}_2^i, \ldots)$	IV of dynamics function, part j-to-i	$\vec{u}_:^{j,i} = (\vec{r}_1^{j,i}, \vec{r}_2^{j,i}, \ldots)$

Index Summary

$\vec{x}_{time}^{part}$	$\vec{u}_{IV-index}^{from-part,to-part}$	$\vec{f}_{time}^{from-part,to-part}$	$\vec{z}_{IP-index}^{from-part,to-part}$
$\vec{g}_{time}^{part}$	$\vec{v}_{IV-index}^{part}$	$\vec{r}_{IP-index}^{part}$	

d-th component of vector $\vec{z}_3^{2,1}$: $(\vec{z}_3^{2,1})_d$

Figure 2. (**Top**): Graphical model representation of the augmented Coupled Gaussian Process Dynamical Model (vCGPDM). Shown is a model with two parts, indicated by the superscripts $i, j \in \{1, 2\}$. Each part is a vGPDM (see Figure 1), augmented with inducing points $\vec{z}_:^{i,j}$ and values $\vec{u}_:^{i,j}$ for variational inference and learning, and modular re-composition of learned GPDM components. Observed variables $\vec{y}_t^i$ and latent-to-observed mappings $\vec{g}^i(\vec{x}_t^i)$ omitted for clarity. The vGPDMs interact by making predictions about each other's latent space evolution via functions $\vec{f}^{i,j}(\vec{x}_{t-2}^i, \vec{x}_{t-1}^i)$, here $\vec{f}^{1,2}()$ and $\vec{f}^{2,1}()$. Their predictions are product-of-experts combined with the predictions made by each GPDM's dynamics model (functions $\vec{f}^{i,i}(\vec{x}_{t-2}^i, \vec{x}_{t-1}^i)$). (**Bottom**): Notation and index summaries.

Besides introducing IPs, computing an ELBO requires a simplifying assumption about the latent state posterior, which is intractable. We choose a posterior distribution q over the latent states $\vec{x}^i_t$ that factorizes across time steps $0, \ldots, T$, parts $1, \ldots, M$ and latent dimensions $1, \ldots, Q^i$ within parts. Furthermore, we assume that the individual distributions are Gaussian:

$$q(\vec{x}^1_0, \ldots, \vec{x}^M_T) = \prod_{t=0}^{T} \prod_{i=1}^{M} \prod_{q=1}^{Q^i} q((\vec{x}^i_t)_q) \ ; \ \ q((\vec{x}^i_t)_q) = \mathcal{N}(\mu^i_{t,q}, \sigma^{2,i}_{t,q}). \tag{3}$$

While this approximation is clearly a gross simplification of the correct latent state posterior, with the right choice of kernels, an ELBO can now be computed. Our approximation assumption (Equation (3)) seems appropriate for human movement data, see Section 5. Whether it is also useful for other data remains to be determined.

As for a tractable kernel, we decided to use an ARD (automatic relevance determination) squared exponential kernel [36] for every part-i-to-j prediction GP:

$$k^{i,j}(\vec{x}^i_{-t}, \vec{x}^{i\,\prime}_{-t}) = \exp\left(-\frac{1}{2} \sum_{t \in -t} \sum_{q}^{Q^i} \frac{((\vec{x}^i_t)_q - (\vec{x}^{i\,\prime}_t)_q)^2}{\lambda^{i,j,t}_q} \right). \tag{4}$$

and a radial basis function kernel for the latent-to-observed mappings. Next, we outline the key steps of the derivation of the ELBO.

4. Computing an Evidence Lower Bound for the vCGPDM: An Overview

In this section, we provide an overview of the derivation of the evidence lower bound (ELBO) for the vCGPDM; for details, the reader is referred to Appendix C. We construct a sparse variational approximation by augmenting each of the $M \times M$ dynamics and coupling mappings $\vec{f}^{i,j}()$ with IPs and IVs. The variational distribution of the latent variables, $q(\vec{x}^{:}_{:}) = q(\vec{x}^1_1, \ldots, \vec{x}^M_T)$ factorizes according to Equation (3). We let $q(\vec{u}^i_{:})$ and $q(\vec{v}^i_{:})$ be unconstrained distributions, which will turn out to be multivariate Gaussians. In the following, we denote the coupling function values at t with $\vec{f}^{i,j}_t = f^{i,j}(\vec{x}^i_{-t})$ and likewise $\vec{g}^i_t = g^i(\vec{x}^i_t)$. The factor structure of the joint density of the augmented model follows from the graphical model (see Figures 1 and 2):

$$p(\vec{y}^{:}_{:}, \vec{g}^{:}_{:}, \vec{v}^{:}_{:}, \vec{x}^{:}_{:}, \vec{f}^{:,:}_{:}, \vec{u}^{:,:}_{:} | \vec{z}^{:,:}_{:}, \vec{r}^{:}_{:}) = p(\vec{y}^{:}_{:}|\vec{g}^{:}_{:}) p(\vec{g}^{:}_{:}|\vec{x}^{:}_{:}, \vec{v}^{:}_{:}, \vec{r}^{:}_{:}) p(\vec{v}^{:}_{:}|\vec{r}^{:}_{:}) p(\vec{x}^{:}_{:}, \vec{f}^{:,:}_{:}|\vec{u}^{:,:}_{:}, \vec{z}^{:,:}_{:}) p(\vec{u}^{:,:}_{:}|\vec{z}^{:,:}_{:}) \tag{5}$$

Note that we marginalized (most of) the functions $f^{:,:}()$ here, keeping only their values at the latent points $\vec{x}^{:}_{:}$ and at the IPs. Hence the dependence of $\vec{g}^{:}_{:}$ on $\vec{r}^{:}_{:}$. Likewise, $\vec{f}^{:,:}_{:}$ depends on $\vec{z}^{:,:}_{:}$. For easier notation, we omit spelling out the dependence of the IVs on the IPs in the following. Thus,

$$p(\vec{y}^{:}_{:}|\vec{g}^{:}_{:}) = \prod_{i=1}^{M} \prod_{d=1}^{D_i} p((\vec{y}^i_{:})_d | (\vec{g}^i_{:})_d) \tag{6}$$

$$p(\vec{g}^{:}_{:}|\vec{x}^{:}_{:}, \vec{v}^{:}_{:}) = \prod_{i=1}^{M} \prod_{d=1}^{D_i} p((\vec{g}^i_{:})_d | \vec{x}^i_{:}, (\vec{v}^i_{:})_d) \tag{7}$$

$$p(\vec{v}^{:}_{:}) = \prod_{i=1}^{M} p(\vec{v}^i_{:}); \ \ p(\vec{u}^{:,:}_{:}) = \prod_{i=1}^{M} \prod_{j=1}^{M} p(\vec{u}^{j,i}_{:}). \tag{8}$$

where Equation (6) follows from the assumption of independent observation noise. Equation (7) is a consequence of the Gaussian process prior on the $g^i()$, which makes the components of $\vec{g}^i_t$ independent. The density of the latent variables and the individual parts' predictions can be factorized as:

$$p(\vec{x}^{:}_{:}, \vec{f}^{:,:}_{:} | \vec{u}^{:,:}_{:}) = p(\vec{x}^{:}_{:} | \vec{f}^{:,:}_{:}, \vec{u}^{:,:}_{:}) p(\vec{f}^{:,:}_{:} | \vec{u}^{:,:}_{:}) \tag{9}$$

with

$$p(\vec{x}^{:}_{:}|\vec{f}^{:,:}_{:},\vec{u}^{:,:}_{:}) = \prod_{t=2}^{T}\prod_{i=1}^{M} p(\vec{x}^{i}_{t}|\vec{f}^{:,i}_{t},\alpha^{:,i}) \tag{10}$$

$$p(\vec{f}^{:,:}_{:}|\vec{u}^{:,:}_{:}) = \prod_{t=2}^{T}\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{f}^{j,i}_{t}|\vec{f}^{j,i}_{1:t-1},\vec{x}^{j}_{0:t-1},\vec{u}^{j,i}_{:}) \tag{11}$$

where Equation (10) follows from the graphical model and the product-of-experts construction (Equation (1)). An empty slice ($t < 2$ for a second-order dynamics model) implies no conditioning. The first two latent states at $t = 0, 1$ are drawn from independent Gaussians, $\prod_{i=1}^{M} p(\vec{x}^{i}_{0})p(\vec{x}^{i}_{1})$. Equation (11) is one possible way of factorizing the augmented Gaussian process prior on the coupling function values: when $\vec{f}^{i,j}()$ is marginalized, the function values at time t depend on all past function values and latent states. We use this particular factorization for analytical convenience. Note that the dependence of the right hand side of Equation (11) on $\vec{x}^{j}_{0:t-1}$ does not contradict the factorization order of Equation (9), because it depends only on latent variables from timesteps prior to t. Furthermore, we choose the following proposal variational posterior:

$$q(\vec{g}^{:}_{:},\vec{x}^{:}_{:},\vec{v}^{:}_{:},,\vec{f}^{:,:}_{:},\vec{u}^{:,:}_{:}) = p(\vec{g}^{:}_{:}|\vec{x}^{:}_{:},\vec{v}^{:}_{:})q(\vec{v}^{:}_{:})p(\vec{f}^{:,:}_{:}|\vec{u}^{:,:}_{:})q(\vec{x}^{:}_{:})q(\vec{u}^{:}_{:}) \tag{12}$$

with $p(\vec{g}^{:}_{:}|\vec{x}^{:}_{:},\vec{v}^{:}_{:})$ given by Equation (7), $p(\vec{f}^{:,:}_{:}|\vec{u}^{:,:}_{:})$ by Equation (11) and $q(\vec{x}^{:}_{:})$ by Equation (3). The densities $q(\vec{v})$ and $q(\vec{u})$ are unconstrained except for normalization. With these distributions, we derive the standard free-energy ELBO [36], denoting $\Theta = (\vec{x}^{:}_{:},\vec{u}^{:,:}_{:},\vec{f}^{:,:}_{:},\vec{v}^{:}_{:},\vec{g}^{:}_{:})$:

$$\log p(\vec{y}^{:}_{:}) \geq \mathcal{L}(\Theta) = \int d\Theta q(\Theta)\log\left(\frac{p(\vec{y}^{:}_{:},\Theta)}{q(\Theta)}\right) \tag{13}$$

exploiting the assumption that the IPs $\vec{r}^{i}_{:}$ and IVs $\vec{v}^{i}_{:}$ are sufficient statistics for the function values $\vec{g}^{i}_{:}$. As we explain in detail in Appendix C, after canceling common factors in the variational posterior (Equation (12)) and the joint model density (Equation (5), cf. [16]), we find that the ELBO can be decomposed into one summand per part that describes the quality of the kinematics mapping (latent-to-observed) $\mathcal{L}^{i}_{kin}$, and one summand for the dynamics $\mathcal{L}_{dyn}$:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{M}\mathcal{L}^{i}_{kin} + \mathcal{L}_{dyn} \tag{14}$$

where

$$\mathcal{L}^{i}_{kin} = \sum_{d=1}^{D}\int d\vec{x}^{i}_{:}\; d(\vec{v}^{i}_{:})_{d}\; d(\vec{g}^{i}_{:})_{d}\; p((\vec{g}^{i}_{:})_{d}|\vec{x}^{i}_{:},(\vec{v}^{i}_{:})_{d})q(\vec{x}^{i}_{:})q((\vec{v}^{i}_{:})_{d})\log\frac{p((\vec{y}^{i}_{:})_{d}|(\vec{g}^{i}_{:})_{d})}{q((\vec{v}^{i}_{:})_{d})}. \tag{15}$$

is—up to the Shannon entropy of approximating posterior of the latent dynamics variables $H(q(\vec{x}^{i}_{:})) = -\int d\vec{x}^{i}_{:}q(\vec{x}^{i}_{:})\log\left(q(\vec{x}^{i}_{:})\right)$—equal to the Bayesian GPLVM ELBO of [16]. The remaining integral

$$\begin{aligned}\mathcal{L}_{dyn} = &\int d\vec{u}^{:,:}_{:}\; q(\vec{u}^{:,:}_{:})\left[\sum_{t=2}^{T}\int d\vec{x}^{:}_{1:t}\; q(\vec{x}^{:}_{1:t})\left(\int d\vec{f}^{:,:}_{t}\; p(\vec{f}^{:,:}_{t}|\vec{f}^{:,:}_{2:t-1},\vec{x}^{:}_{0:t-1},\vec{u}^{:,:}_{:})\log p(\vec{x}^{:}_{t}|\vec{f}^{:,:}_{t},\alpha^{:,:})\right)\right]\\ &+\int d\vec{u}^{:,:}_{:}q(\vec{u}^{:,:}_{:})\log\frac{p(\vec{u}^{:,:}_{:})}{q(\vec{u}^{:,:}_{:})}+\int d\vec{x}^{:}_{0:1}q(\vec{x}^{:}_{0:1})\log p(\vec{x}^{:}_{0:1})+H(q(\vec{x}^{:}_{:}))\end{aligned} \tag{16}$$

is derived in detail in Appendix C. Briefly, we use the assumption that the IPs and IVs $\vec{z}^{j,i}_{:}$ and $\vec{u}^{j,i}_{:}$ are sufficient statistics for the function values $\vec{f}^{j,i}_{t}$. Optimizing with respect to $q(\vec{u}^{:,:}\; :)$ can be carried out in closed form using variational calculus and yields

$$\mathcal{L}_{dyn}(\Theta) \geq \log \int p(\vec{u}_{:}^{:,:}) \exp(\mathcal{C}(\vec{u}_{:}^{:,:})) d\vec{u}_{:}^{:,:} + H(q(\vec{x}_{:}^{:})) \tag{17}$$

where $\mathcal{C}(\vec{u}_{:}^{:,:})$ is given by Equation (A31). The inequality is due to the sufficient statistics assumption, which introduces another approximation step that lower-bounds $\mathcal{L}_{dyn}$. We now have all the ingredients to compute the ELBO for the whole model, and learn it.

5. Results

We used the machine-learning framework `Theano` [37] for automatic differentiation in `Python 2.7` (Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org) to implement the model, and learned via optimization of the ELBO with the `scipy.optimize.fmin_l_bfgs_b` routine [38]. Latent space trajectories were initialized with PCA. We obtained the best ELBOs by first optimizing all parameters jointly, followed by a blocked optimization procedure. We optimize three groups of parameters: latent points and variances, kernel parameters and couplings, and IPs. The number of iterations of the blocked procedure depended on the application; we provide details in the sections below.

The advantage of the sparse approximations in the vCGPDM is that memory consumption of the model is greatly reduced. However, this approximation might also introduce errors, along with the fully factorized latent posterior (Equation (3)). We tried to quantify these errors in a cross-validatory model comparison, and in a human perception experiment.

5.1. Synthetic Data

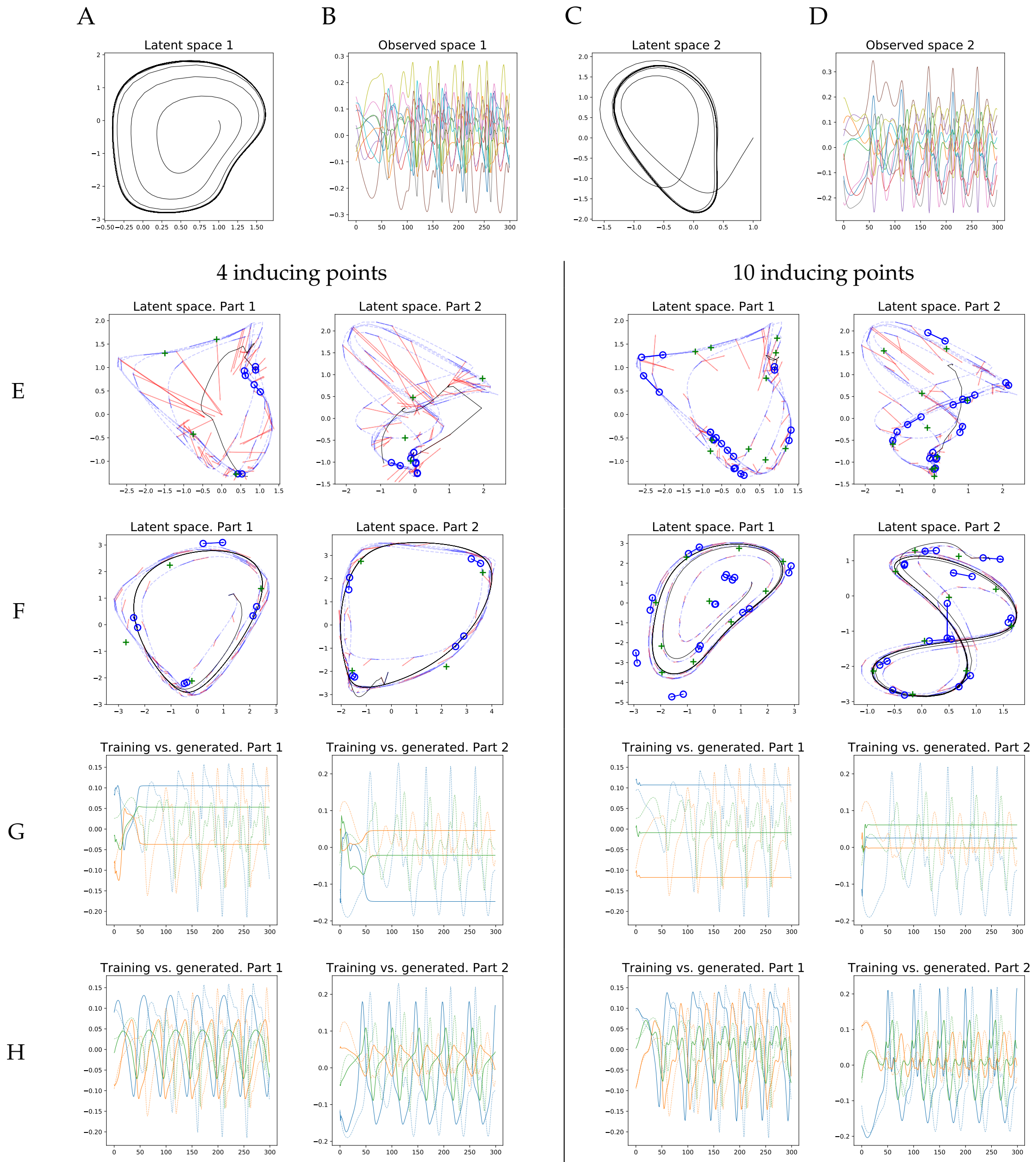
We demonstrate the learning of coupled dynamical systems on a synthetic dataset. First, we draw two dynamics transition functions $g^1, g^2 \in \mathbb{R}^2 \rightarrow \mathbb{R}^2$ from a $\mathcal{GP}$ with an RBF kernel, and then we generate latent trajectories according to:

$$\vec{x}_t^1 = g^1(\vec{x}_{t-1}^1) \tag{18}$$

$$\vec{x}_t^2 = 0.1g^1(\vec{x}_{t-1}^1) + 0.9g^2(\vec{x}_{t-1}^2) \tag{19}$$

at $T = 300$ timepoints. Thus, we get two first-order, coupled latent dynamical systems, each of dimensionality 2. The trajectory in Latent Space 1 is independent of Latent Space 2, whereas Latent Space 2 is weakly coupled to Latent Space 1. Then, for each of the two parts, we draw 10 observed trajectories from another two RBF GPs with inputs on the latent trajectories. The latent trajectories are shown in Figure 3A,C. Figure 3B,D displays the corresponding observed trajectories. We learned a second-order vCGPDM from these data, iterating the blocked optimization until convergence of the ELBO to machine precision. We chose a second order system for this learning example, because the human movement models in the following are second-order vCGPDMs, too.

The results are shown in Figure 3E–H. Plots on the left half were generated with four IPs, plots in the right half with ten IPs. Figure 3E displays the initial positions of the dynamics IPs (blue circles) at the beginning of learning, connected circles form one second-order IP. Green crosses are the kinematics IPs (latent-to-observed mapping). Initial latent points (dashed blue lines) were obtained from the first two PCA components of the training data. Blue and red line segments are examples of the dynamics mapping: the end-points of the blue segments are the inputs, the distal endpoint of the red segment is the output. As one might expect, the initial conditions do not describe a dynamics which can produce trajectories resembling those in the training data: the black line is the mean latent trajectory, and Figure 3G shows the corresponding observable trajectories.



$\alpha^{i,j}$	$j=1$	$j=2$
$i=1$	$\alpha^{1,1}=0.092$	$\alpha^{1,2}=0.55$
$i=2$	$\alpha^{2,1}=4.9$	$\alpha^{2,2}=0.27$
	initial	final
ELBO:	$-9.0\cdot10^{3}$	$1.0\cdot10^{4}$
MSE:	$4.7\cdot10^{-3}$	$1.3\cdot10^{-3}$

$\alpha^{i,j}$	$j=1$	$j=2$
$i=1$	$\alpha^{1,1}=9.6\cdot10^{-3}$	$\alpha^{1,2}=0.023$
$i=2$	$\alpha^{2,1}=10$	$\alpha^{2,2}=0.011$
	initial	final
ELBO:	$-8.1\cdot10^{3}$	$1.3\cdot10^{4}$
MSE:	$5.7\cdot10^{-3}$	$6.1\cdot10^{-4}$

Figure 3. Synthetic training data. (**A**–**D**) Two-dimensional latent dynamics trajectories and corresponding observed 10-dimensional time series. Part 2 is weakly influenced by Part 1, Part 1 is not influenced by Part 2 (see Equation (18)). (**E**) Initial positions of second-order dynamics IPs (connected blue circles) and latent-to-observed IPs (green crosses). Line segments are examples of dynamics mapping inputs (endpoints of blue segments) and values (distal endpoints of red line segments). Black line: mean trajectory, generated by iterating the mean dynamics mapping from the same starting point as in (**F**) Latent space after learning. (**G**) Generated observable time series before learning (solid) and training data (dashed). (**H**) Generated time series (solid) after learning. Only three of the ten observable trajectories are presented for clarity. (**Bottom**) Learned couplings (see Equation (1)); ELBOs and MSEs rounded to two significant digits. Couplings $\alpha^{i,j}$ reflect the dependency structure between parts: Part 1 is not driven by Part 2, but influences Part 2.

After learning, the latent trajectories appear to have a limit cycle (black line, Figure 3F), which is required to reproduce the training data. Furthermore, note that the inducing points align with that cycle, and the example mappings (blue and red line segments) indicate clearly how the latent trajectory is formed by iterating the GP mapping. Cross coupling GP mappings omitted for clarity. Note that the vCGPDM with ten IPs can model more complex latent dynamics manifolds than the four-IP vCGPDM. The observable trajectories (Figure 3H) look very similar to the training data up to a small phase shift, particularly for the 10 IP model. This observation is confirmed by the reduced mean squared trajectory error (MSE) between generated and training data after learning, which was evaluated after dynamic time warping [39] of the generated trajectories onto the training data. The MSEs are listed in the table at the bottom of Figure 3, where "final" indicates the values after learning, while "initial" indicates the values at the onset of learning after the latent space trajectories had been initialized to the first two PCA components of the training data. That learning was successful is also indicated by the increased final ELBO, which is higher for the 10 IP model.

We also provided the learned coupling αs in this table. Recall that a low (high) α means a large (small) influence of the corresponding part on the dynamics. The dependency structure between the latent spaces was correctly identified during learning: $\alpha^{2,1} \gg \alpha^{1,1}$, i.e., Part 2 has almost no influence on Part 1. In contrast, $\alpha^{1,2} \approx 2\alpha^{2,2}$, which indicates that Part 1 weakly controls Part 2.

5.2. Human Movement Data

Model comparisons and psychophysical tests were carried out on human movement data. We employed a 10-camera PhaseSpace Impulse motion capture system, mapped the resulting position data onto a skeleton with 19 joints and computed joint angles in exponential-map representation, yielding a total of 60 degrees of freedom. Five walking-only and four walking + waving sequences each were used to train the models, as well as ten movements where the human participants were seated and passed a bottle from one hand to the other. Dynamical models were initialized with starting conditions taken from the training data. The blocked optimization was run for at most four iterations, which was enough to ensure convergence. It was terminated earlier if ELBO values did not change within machine precision between two subsequent iterations. Furthermore, we recorded another nine walking sequences for catch trials during the perception experiment, to rule out memorization effects. Generated and recorded sequences were rendered on a neutral avatar. Examples of stimuli, for different numbers of IPs, can be found in the movie `example_stimuli.mov` in the Supplementary Materials.

5.3. Variational Approximations are Better than MAP

We performed cross-validatory model comparison on the following datasets: walking, walking + waving and passing-a-bottle. Examples of these data are shown in the movies in the Supplementary Materials: `S1_example_stimuli.mov` and `S4_pass_the_bottle.mkv`. We performed four-, five- and ten-fold crossvalidation, the number of folds was dictated by the dataset size. We were trying determine how the sparsely parameterized vCGPDM performs in comparison to the full CGPDM, and several other MP models from the literature. Held-out data were always one complete trial. Models were trained on the remaining data and the generated trajectory was compared to the held-out one. Cross-validation score was the mean-squared error (MSE) of the kinematics after dynamic time warping [39] of trajectories generated by initializing the model to the first two frames of a held-out trial onto the complete held-out trial. We used dynamic time warping to compensate a slight phase difference in generated motions, which would otherwise lead to a much larger and uninformative MSE. We compared the following models:

- a GPDM with maximum-a-posteriori (MAP) estimation of the latent variables [25], called MAP GPDM in Figure 4;
- a fully marginalized two-part (upper/lower body) CGPDM with MAP estimation of the latent variables [14], called MAP CGPDM U+L;

- a three-part CGPDM model (left hand, right hand, and body) for the non-periodic "passing a bottle" dataset;
- their variational counterparts, vCGPDM 3-part, vCGPDM U+L and vGPDM;
- temporal MPs (TMP, instantaneous linear mixtures of functions of time) [9]; and
- DMPs [12].

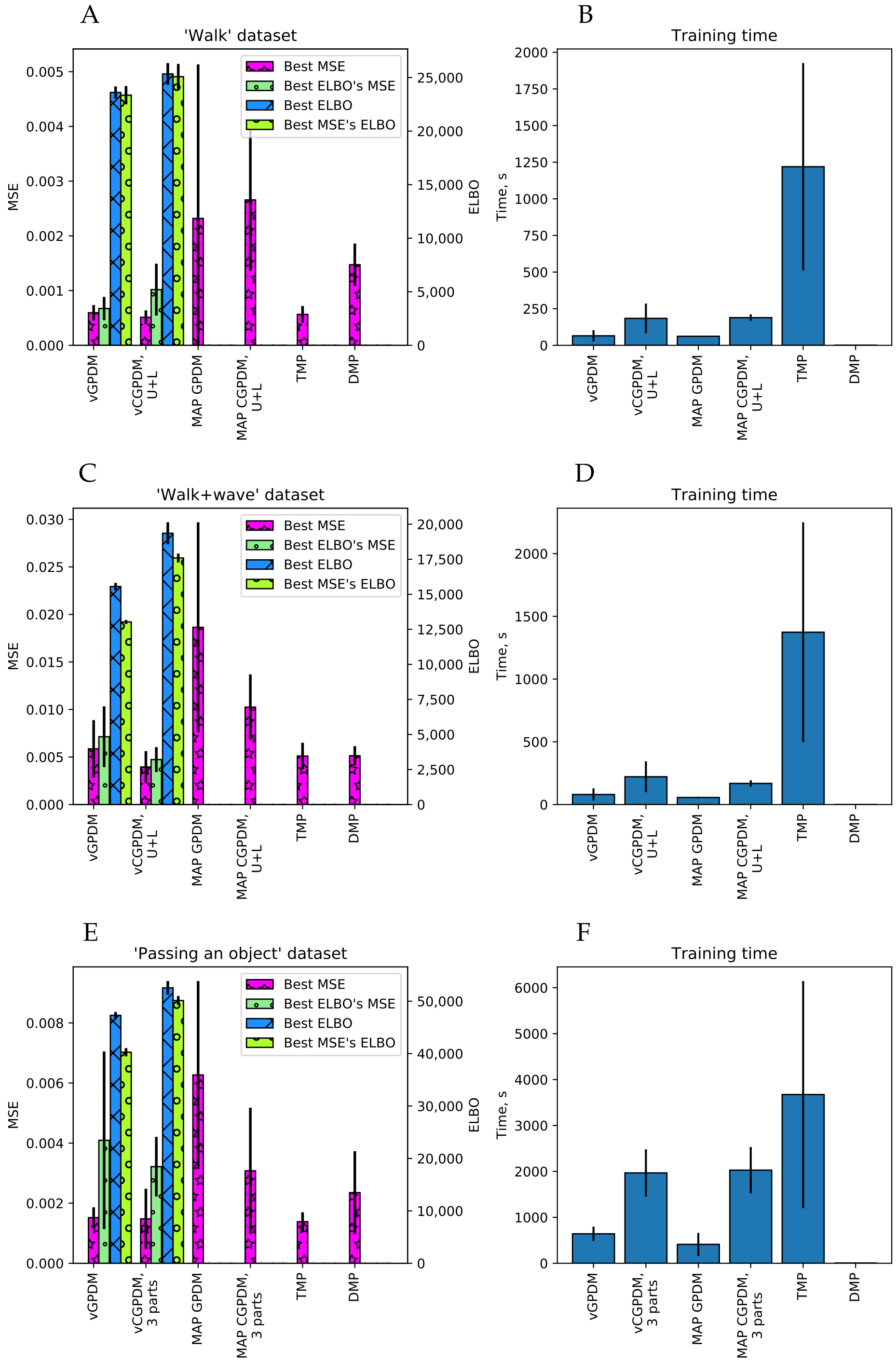


Figure 4. Model comparison results. We plotted the average squared kinematics error on held-out data after dynamic time warping (MSE) and the variational lower bound on the model evidence (ELBO, Equation (A22)), where available, accompanied with corresponding model training time. Error bars are standard errors of the mean. (**A**,**B**) Walking dataset; (**C**,**D**) walking + waving dataset; and (**E**,**F**) "passing a bottle" dataset. Low MSE and high ELBO are better. For details, see text. Figure partially adapted from [19].

All latent spaces were three-dimensional. We tried 4–30 latent-to-observed IPs and 2–30 dynamics IPs. The MSE optima were near 10–15 IPs for both the walking and the walking + waving datasets, and near eight IPs for the "passing a bottle". MAP GPDM and MAP CGPDM learning do not use any approximations or inducing points; they are the full GPs with covariance matrices $\mathbf{K} \in \mathbb{R}^{T*T}$.

For the TMPs, we used up to 10 primitives; the MSE optimum was located at approximately six. For the DMPs, we used between 1 and 50 basis functions, and the lowest MSE was found around 15.

The results are plotted in Figure 4. Generally, the walking + waving movement is more difficult to reproduce for all models than walking only: the MSE of the latter is lower than that of the former, and the ELBO is higher. This indicates that the latter is a more complex movement, see also the movie `modular_primitives.avi` in the online Supplementary Materials. The two-part vCGPDM reaches the lowest MSE compared to all other models. Clearly, it is better than the full-capacity (no IPs) MAP models, which means that the extra effort of developing of a variational approximation which explicitly represents an approximation to the latent states' posterior and needs to store only ≈10 IPs rather than $\approx 10^4$ data points was well spent. In addition, the best ELBO's MSE (that is, the MSE at the maximum of the ELBO) is a fairly good predictor of the best MSE, which justifies our simple variational approximation for model selection.

The vCGPDM U+L outperforms the vGPDM particularly on the "walking + waving" dataset. This shows the usefulness of having modular, coupled dynamics models when the (inter)acting (body)parts execute partially independent movements.

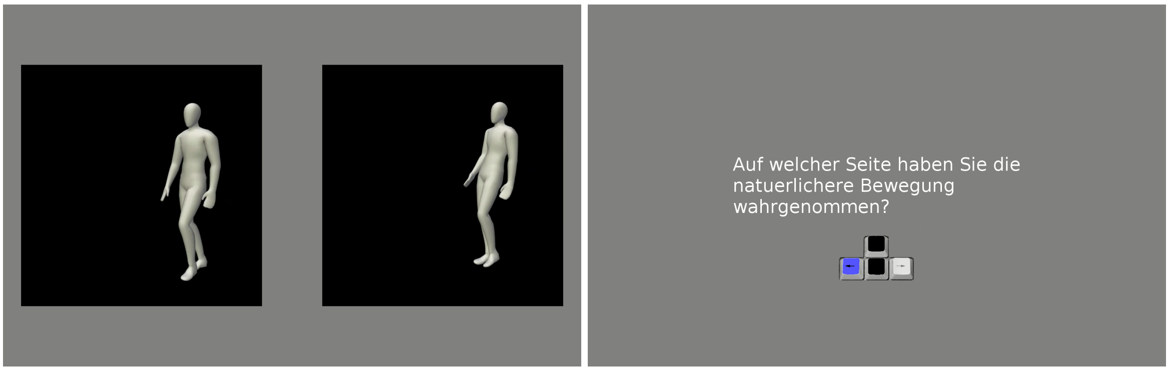
The vCGPDM with three parts for "passing a bottle" does not show a clear advantage over the monolithic model in the cross-validation test, and is on par with the TMP model. However, dynamics factorization did not affect the performance either. This may be indicative for the strong coupling necessary to successfully pass an object from one hand to the other. Such a strong coupling is parsimoniously expressed by having a single latent dynamical system drive all observable degrees of freedom.

The timing results show that the training times of the vCGPDM are usually less than 15 min. Error bars are standard deviations, estimated across all numbers of IPs and cross-validation splits. The rather large training time for the TMP model is due to the implementation from [9] which optimizes a rather large covariance matrix between all MPs.

5.4. A Small Number of IPs Yields Perceptually Convincing Movements

We conducted a psychophysical experiment to quantify the perceptual validity of the generated movements. More specifically, we investigated the model complexity required for perceptually convincing movements.

Experiment: Thirty-one human observers (10 male, mean age: 23.8 ± 3.5a) participated in a two-alternative forced-choice task to distinguish between natural and generated movements (see Figure 5 for an example of an experimental trial). Natural movements consisted of 15 walking movements. The artificial movements were generated by a two-part (upper/lower body) vCGPDM. We used 2–16 dynamics IPs and 4–16 latent-to-observed IPs. We chose these numbers based on pilot tests to span the range from clearly unnatural to very natural looking movements. To test whether participants simply memorized the 15 natural stimuli during the experiment, we added 10 catch trials in the last quarter of the experiment where previously unused natural movements were tested against the known natural stimuli. The trial sequence was randomized for every subject. All experimental procedures were approved by the local ethics commission.



Auf welcher Seite haben Sie die natuerlichere Bewegung wahrgenommen?

Figure 5. Psychophysical Experiment. In each trial, a natural and a generated movement were simultaneously presented to participants (**left**). After presentation, they used the arrow keys to choose the movement perceived as more natural (**right**). There was no time limit on the response, but typically participants responded quickly (less than 1 s). After the response, people were asked to fixate a cross in the middle of the screen, which appeared for a second. The length of the stimuli was 1.8 s, with a total of 1170 presentations. A video of the experiment called `S2_experiment_demo.avi` is provided in the Supplementary Materials.

Results: We computed the confusion rate, i.e., the frequency of choosing the model-generated movement as more natural across all participants as a function of the number of IPs for the dynamics and latent-to-observed mappings. Optimally, we might expect this rate to approach $\frac{1}{2}$ when the generated movements are indistinguishable from the natural ones. We investigated if the confusion rate approached this limit, how it depends on the mean-squared residual error on the training data, and how this error is connected to the ELBO. The results are plotted in Figure 6. We also fitted the confusion rate data with a logistic sigmoid $\frac{0.5}{1+\exp(a \cdot MSE+c)}$ (solid line in Figure 6A), and the MSE with an exponential function (solid line in Figure 6, right). Each data point represents one combination of dynamics/latent-to-observed IP numbers, indicated by width and height of the ellipses. Clearly, confusion rate increases fairly monotonically with decreasing MSE, as indicated by the good logistic sigmoid fit. Furthermore, models with more IPs also tend to yield higher confusion rates. A sufficient number (>10) dynamics IPs is more important than a large number of latent-to-observed IPs, which can be seen by the very narrow ellipses in the region with high MSE, and many wider ellipses in the lower MSE part of the figure. A similar observation can be made about the relationship between ELBO and MSE (Figure 6B). It indicates that ELBO is already a good predictor for the model performance. For a very small number of dynamics IPs, increasing the number of latent-to-observed IPs does not decrease the MSE as much as increasing the dynamics IPs does. Moreover, note that the relationship between MSE and ELBO becomes fairly monotonic when ELBO > 28,500, which is where human perceptual performance can be predicted from ELBO. While the confusion rate has not quite reached its theoretical maximum in our experiment, these results are evidence that human perceptual expectations can be nearly met with very compactly parameterized MP models (Figure 6C,D). Moreover, good dynamics models seem to outweigh precise kinematics. We found no evidence for stimulus memorization from the confusion rates of the catch trials.

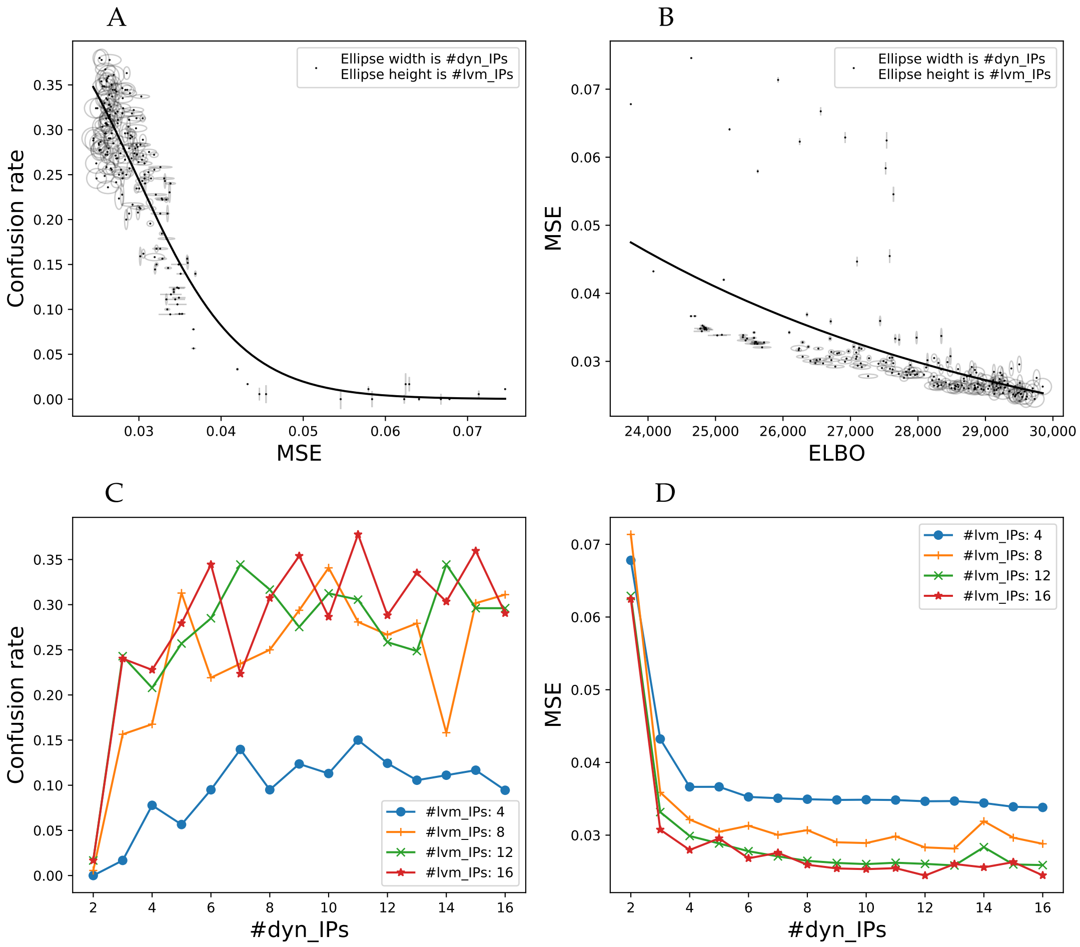
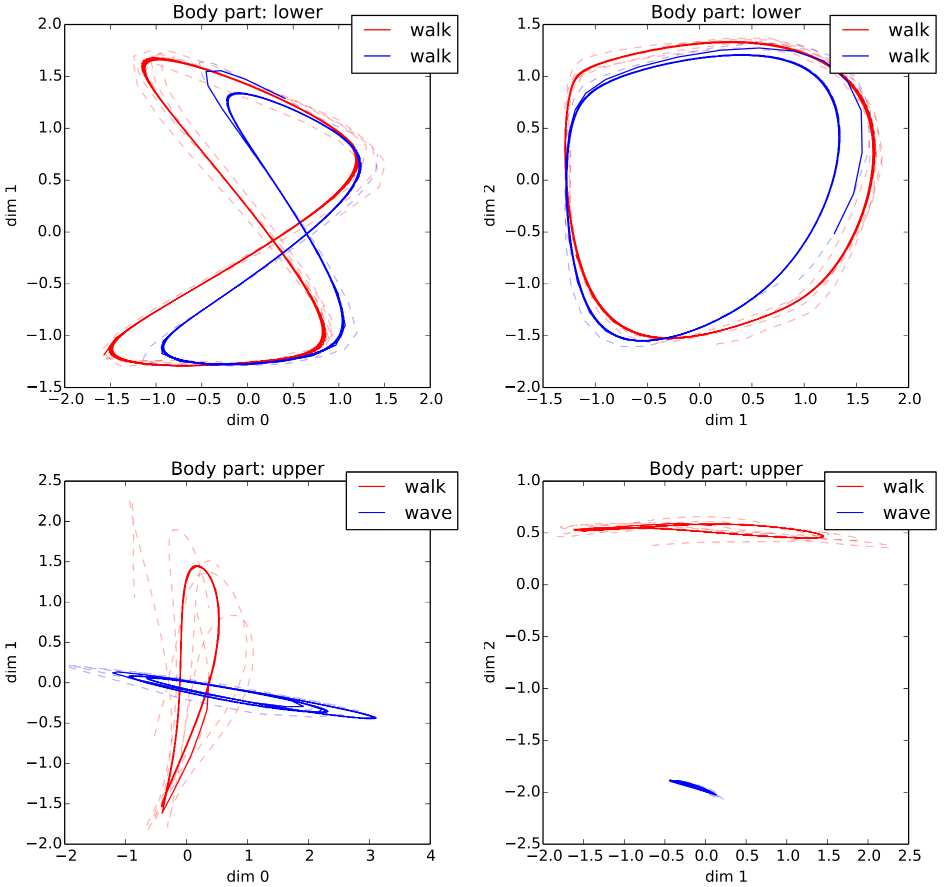


Figure 6. (**A**) Confusion rate between natural and vCGPDM-generated stimuli as a function of mean-squared residual error (MSE) on the training data, averaged across all participants. Each data point represents one combination between number of IPs/IVs for the latent-to-observed mapping (indicated by ellipse height) and number of IPs/IVs for the dynamics mappings (ellipse widths). A confusion rate of 0.5 indicates that human observers are not able to distinguish replays of real movements from model-generated counterparts. The vCGPDM is approaching this limit from below for a fairly small number of IPs/IVs. Solid line: fit with logistic sigmoid function. (**B**) Relationship between training MSE and ELBO. Solid line: fit with exponential function. Additional dynamics IPs contribute more to the reduction of the MSE than latent-to-observed IPs. MSE and therefore confusion rate can be predicted well from ELBO if ELBO > 28,500. (**C**,**D**) Influence of number of dynamics IPs on the confusion rate and MSE, respectively, for a selected number of latent-to-observed IPs. The confusion rate has a broad maximum around 8–12 dynamics IPs, whereas the MSE has a shallow minimum at that location.

5.5. Modularity Test

Next, we examined if the intended modularization of our model can be used to compose novel movements from previously learned parts. We trained a vCGPDM consisting of one part for the lower body (below and including pelvis), and a second part for the upper body. Twenty-five IPs for the latent-to-observed mapping of each part were shared across all movements. The walking MP, parameterized by 16 IPs for the lower-body dynamics and the lower-to-upper mappings, was also shared. We used a different set of 16 IPs for the upper body MPs between arm-swing and waving. Furthermore, the coupling $\alpha^{j,i}$ were learned anew for each combination of upper/lower MPs. The resulting latent space trajectories are plotted in Figure 7. All generated trajectories (solid lines) are on average close to the training data (dashed lines). While the walking trajectories for the lower body are very similar for the two movements, the upper body trajectories clearly differ. Movements generated from this model are very natural (see video `S3_modular_primitives.mov` in the Supplementary Materials). This is a first demonstration that the vCGPDM with non-marginalized couplings can be used to learn a library of compactly parameterized MPs, from which novel movements can be produced with little additional memory requirements (i.e., new coupling $\alpha^{j,i}$ only).



Dataset		vCGPDM	Modular vCGPDM
Walking	Cross-validation MSE	0.0042(±0.0035)	0.0042(±0.0017)
	Training time, s	1948(±225)	6289(±709)*
Walking and waving	Cross-validation MSE	0.0050(±0.0006)	0.0057(±0.0013)
	Training time, s	1501(±236)	6289(±709)*

Figure 7. (**Top**) Modularity example. Shown are 2D projections of generated 3D latent space trajectories (solid) and training data (dashed). Blue: walk + wave movements; red: walk + normal arm swing. Dynamics IPs re-used across movements for lower body. (**Bottom**) Cross-validation MSEs of non-modular and modular vCGPDM. Modular vCGPDM was trained on the combined dataset; training time (*) is shared between both movement datasets.

For a quantitative evaluation, we looked at the leave-one-out cross-validation MSEs of the vCGPDM trained on datasets separately and modular vCGPDM trained on both datasets (see Figure 7, bottom). Within the standard errors, MSEs are equal, indicating that modular re-use of previously trained components does not necessarily sacrifice accuracy, while reducing storage requirements. Training time for the modular vCGPDM is larger due to the learning of the combined dataset and optimizing the couplings afterwards. This time would be amortized if more compositional movements were learned, where previously learned parts could be reused.

6. Conclusions

The vCGPDM, a full variational approximation to the CGPDM, allows for learning a deterministic approximation of latent space trajectories, and compactly parameterizing dynamics and kinematics mappings. First, we showed that the sparsely parameterized vCGPDM outperforms the full-capacity, monolithic CGPDM employing MAP to approximate the latent dynamics posterior. It also surpasses other current MP models; we speculate that this is accomplished by its learnable dynamics.

Second, we demonstrated that our compact representation of the latent space dynamics, and of the latent-to-observed mapping, enables the model to generate perceptually convincing full-body movements with a fairly small number of IPs To our knowledge, a systematic investigation of the number of IPs needed for perceptual plausibility had not been done before, albeit more monolithic models were in the focus of earlier studies [27,34,40]. Moreover, we demonstrated that a high enough ELBO can be used to predict average human classification performance, which might allow for an automatic model selection process when training the model on large databases. Within the range of IPs which we tested, the ELBO was still increasing with their number. We chose that range because we wanted to see how few IPs would still lead to perceptually indistinguishable movements. Due to experimental time constraints, we did not investigate perceptual performance at the point where the ELBO begins to decrease with increasing IPs (i.e., the approximately optimal model), but we plan to do that in the future.

Third, we showed that the model can be employed in a modular fashion, using one lower-body dynamics model, and coupling it to two different models for the upper body. Note that the lower-to-upper coupling function was the same for the two upper-body models. Each of these models, including the coupling functions to the other model parts, may therefore be viewed as a modular MP that is parameterized compactly by a small number of IPs and values. This sparse parameterization allows us to infer modular MPs from a large collection of movements, and investigate their composition. To generate complex movement sequences, we will put a switching prior on top of the dynamical models, as in [29].

We are currently researching sensorimotor primitives, i.e., MPs that can be used to predict sensory input and be controlled by it via conditioning. This conditioning can take place on at least two timescales: a short one (while the MP is running), thus effectively turning the MPs into flexible control policies, such the probabilistic MPs described by Paraschos et al. [41], and a long timescale, i.e., the planning of the movement. This could be implemented by learning a mapping from goals and affordances onto the coupling weights, comparable to the DMPs with associative skill memories [42]. There is evidence that humans modulate the coupling between their MPs during the planning stage: whole-body posture changes have been observed in anticipation of reaching for a goal object in a known location, even if the object is currently invisible [43].

Lastly, we note that our CGPDM could be used as a flexible policy model for PILCO-style reinforcement learning (Probabilistic Inference for Learning Control, [44]). PILCO requires a dynamics model that can propagate uncertainties through time; the vCGPDM is able to do that. Thus, our model could be used as a lower dimensional dynamics model which can capture the dependencies between observable variables via latent space uncertainties.

Supplementary Materials: The following are available online http://www.mdpi.com/1099-4300/20/10/724/s1, Video S1: example_stimuli, Video S2: experiment_demo, Video S3: modular_primitives, Video S4: pass_the_bottle.

Author Contributions: D.V. and D.E. conceived vCGPDM model. D.V. derived, implemented and tested models. B.K. and D.E. conceived psychophysical experiment. B.K. conducted experiment and evaluated data. All authors wrote paper.

Funding: This work was supported by the Deutsche Forschungsgemeinschaft under DFG-IRTG 1901 “The Brain in Action” and DFG-SFB-TRR 135 project C06.

Acknowledgments: We thank Olaf Haag for help with rendering the movies, and Björn Büdenbender for assistance with MoCap. Open access costs were paid by the University Library of Marburg and the DFG.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MP	movement primitive
DMP	dynamical movement primitive
$\mathcal{GP}$	Gaussian process
GPDM	Gaussian process dynamical model
CGPDM	coupled Gaussian process dynamical model
vCGPDM	variational coupled Gaussian process dynamical model
IP	inducing point
IV	inducing value
MSE	mean squared error

Appendix A. Exact Variational Optimization of Parts of the ELBO

While optimizing the full variational posterior in augmented Gaussian processes models with respect to the IVs, the following type of term appears several times in the ELBO equation:

$$\begin{aligned}\mathcal{R}(q(\vec{u}), r(\vec{v})) &= \int q(\vec{u}) \left(f(r(\vec{v}), \vec{u}) + \log \tfrac{p(\vec{u})}{q(\vec{u})} \right) d\vec{u} \\ &= \int q(\vec{u}) f(r(\vec{v}), \vec{u}) d\vec{u} + \int q(\vec{u}) \log p(\vec{u}) d\vec{u} - \int q(\vec{u}) \log q(\vec{u}) d\vec{u}\end{aligned} \tag{A1}$$

To simplify the optimization of such terms, we would like to carry out the optimization with respect to the density $q(\vec{u})$ analytically to remove the dependency on $q(\vec{u})$. Note that we allow only $q(\vec{u}), r(\vec{v})$ to vary, while the functions $f(r(\vec{v}), \vec{u})$ and $p(\vec{u})$ are assumed to be fixed. To this end, we calculate for the optimal variational $q^*(\vec{u})$ in the above equation. This approach was suggested in [23], however, it is not well described there. Here, we give an extended derivation. A necessary condition for maximality is a vanishing functional derivative under the constraint that the density $q(\vec{u})$ is normalized to one:

$$\int q(\vec{u}) d\vec{u} - 1 = 0 \tag{A2}$$

which is fulfilled at the stationary points of the Lagrangian

$$\mathcal{X}(q(\vec{u}), r(\vec{v})) = \mathcal{R}(q(\vec{u}), r(\vec{v})) + \lambda \left(\int q(\vec{u}) d\vec{u} - 1 \right) \tag{A3}$$

where λ is chosen so that Equation (A2) holds. Taking the functional derivative of $\mathcal{X}(q(\vec{u}), q(\vec{v}))$ and setting it to zero yields

$$\frac{\delta \mathcal{X}(q(\vec{u}), q(\vec{v}))}{\delta q(\vec{u})} = f(r(\vec{v}), \vec{u}) + \log p(\vec{u}) - \log q(\vec{u}) - 1 + \lambda = 0 \tag{A4}$$

and therefore, denoting $Z = \exp(-\lambda + 1)$

$$q^*(\vec{u}) = \exp(f(r(\vec{v}), \vec{u}) + \log p(\vec{u}) - 1 + \lambda) \tag{A5}$$

$$q^*(\vec{u}) = \frac{1}{Z} p(\vec{u}) \exp(f(r(\vec{v}), \vec{u})) \tag{A6}$$

$$Z = \exp(-\lambda + 1) = \int p(\vec{u}) \exp(f(r(\vec{v}), \vec{u})) d\vec{u} \tag{A7}$$

Substituting the optimal $q^*(\vec{u})$ into the original term, we get:

$$\begin{aligned}\mathcal{R}(r(\vec{v})) &= \int \tfrac{1}{Z}p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))\left(f(r(\vec{v}),\vec{u}) + \log\frac{p(\vec{u})}{\frac{1}{Z}p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))}\right)d\vec{u}\\ &= \int \tfrac{1}{Z}p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))\left(\log\frac{p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))}{\frac{1}{Z}p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))}\right)d\vec{u}\\ &= \log(Z)\tfrac{1}{Z}\int p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))d\vec{u}\\ &= \log\int p(\vec{u})\exp(f(r(\vec{v}),\vec{u}))d\vec{u}\end{aligned} \tag{A8}$$

This is the optimized version of Equation (A1), which depends only on $r(\vec{v})$.

Appendix B. ARD RBF Kernel Ψ Statistics. Full Covariance Variational Parameters Case.

During the computation of the ELBO, it is necessary to evaluate expected values of Gaussian process kernel functions under the variational posterior distributions. Here, we derive these expectations, referred to as Ψ statistics in the literature [16], for the type of kernel we used in this paper: an automatic relevance determination, squared exponential kernel (ARD RBF). The ARD RBF kernel is defined as:

$$k(\vec{x},\vec{x}') = \sigma_f^2\exp\left(-\frac{1}{2}\sum_{q=1}^{Q}\frac{((\vec{x})_q - (\vec{x})'_q)^2}{\lambda_q}\right) \tag{A9}$$

where λ_q are the ARD factors, σ_f^2 is the variance of the kernel and Q is the dimensionality of $\vec{x}$. In matrix notation:

$$\lambda = \mathrm{diag}(\lambda_1\ldots\lambda_Q) \tag{A10}$$

$$k(\vec{x},\vec{x}') = \sigma_f^2\exp\left(-\frac{1}{2}(\vec{x}-\vec{x}')^T\lambda^{-1}(\vec{x}-\vec{x}')\right) \tag{A11}$$

Let $\vec{x}$ be a random variable drawn from multivariate Gaussian distributions with mean $\vec{\mu}$ and covariance matrix S. Consider the following form of the approximate variational posterior distribution of $\vec{x}$:

$$q(\vec{x}) = \mathcal{N}(\vec{x}_n|\vec{\mu},S) \tag{A12}$$

The Ψ_0 statistic is the expectation of the kernel for two identical arguments, which is easy to calculate:

$$\begin{aligned}\Psi_0 &= \int k(\vec{x}_n,\vec{x}_n)\mathcal{N}(\vec{x}_n|\vec{\mu}_n,S_n)d\vec{x}_n\\ &= \int \sigma_f^2\mathcal{N}(\vec{x}_n|\vec{\mu}_n,S_n)d\vec{x}_n\\ &= \sigma_f^2\int\mathcal{N}(\vec{x}_n|\vec{\mu}_n,S_n)d\vec{x}_n\\ &= \sigma_f^2\end{aligned} \tag{A13}$$

The Ψ_1 statistic is the expectation with respect to one kernel argument, given that the other is constant:

$$\begin{aligned}\Psi_1 &= \int k(\vec{x},\vec{z})\mathcal{N}(\vec{x}|\vec{\mu},S)d\vec{x}\\ &= \int \sigma_f^2\exp\left(-\tfrac{1}{2}(\vec{x}-\vec{z})^T\lambda^{-1}(\vec{x}-\vec{z})\right)\mathcal{N}(\vec{x}|\vec{\mu},S)d\vec{x}\end{aligned} \tag{A14}$$

To evaluate the integral, complete the ARD RBF kernel to a scaled Gaussian distribution with covariance matrix λ and mean $\vec{z}$:

$$\begin{aligned}\Psi_1 &= \sigma_f^2\int\tfrac{Z(\lambda)}{Z(\lambda)}\exp\left(-\tfrac{1}{2}(\vec{x}_r-\vec{z})^T\lambda^{-1}(\vec{x}_r-\vec{z})\right)\mathcal{N}(\vec{x}_n|\vec{\mu}_n,S_n)d\vec{x}\\ &= \sigma_f^2\,Z(\lambda)\int\mathcal{N}(\vec{x}|\vec{z},\lambda)\mathcal{N}(\vec{x}|\vec{\mu},S)d\vec{x}\end{aligned} \tag{A15}$$

with $Z(\lambda) = (2\pi)^{Q/2}\sqrt{|\lambda|}$. The integral over the product of two Gaussians can be carried out to yield (see [45], Identity 371):

$$\begin{aligned}
\Psi_1 &= \sigma_f^2\, Z(\lambda)\mathcal{N}(\vec{z}|\vec{\mu}, \lambda + S) \\
&= \sigma_f^2\, (2\pi)^{Q/2}\sqrt{|\lambda|}\frac{1}{(2\pi)^{Q/2}\sqrt{|\lambda + S|}} \exp\left(-\tfrac{1}{2}(\vec{z} - \vec{\mu})^T(\lambda + S)^{-1}(\vec{z} - \vec{\mu})\right) \\
&= \sigma_f^2\, \frac{\sqrt{|\lambda|}}{\sqrt{|\lambda + S|}} \exp\left(-\tfrac{1}{2}(\vec{z} - \vec{\mu})^T(\lambda + S)^{-1}(\vec{z} - \vec{\mu})\right) \\
&= \sigma_f^2\, \sqrt{\frac{\prod_{q=1}^{Q}\lambda_q}{|\lambda + S|}} \exp\left(-\tfrac{1}{2}(\vec{z} - \vec{\mu})^T(\lambda + S)^{-1}(\vec{z} - \vec{\mu})\right)
\end{aligned} \tag{A16}$$

The Ψ_2 statistic integral, which correlates two kernel function values at different points $\vec{z}$ and $\vec{z}'$, can be solved in a similar manner: first, by collecting terms in the exponents and completing quadratic forms, and second, by the application of Identity 371 from [45]:

$$\begin{aligned}
\Psi_2 &= \int k(\vec{x}, \vec{z})k(\vec{z}', \vec{x})\mathcal{N}(\vec{x}|\vec{\mu}, S)d\vec{x} \\
&= (\sigma_f^2\, Z(\lambda))^2\ \int \mathcal{N}(\vec{x}|\vec{z}, \lambda)\mathcal{N}(\vec{x}|\vec{z}', \lambda)\mathcal{N}(\vec{x}|\vec{\mu}, S)d\vec{x} \\
&= (\sigma_f^2 Z(\lambda))^2\ \int \mathcal{N}(\vec{z}'|\vec{z}, 2\lambda)\mathcal{N}(\vec{x}|\frac{1}{2}(\vec{z}' + \vec{z}), \frac{1}{2}\lambda)\mathcal{N}(\vec{x}|\vec{\mu}, S)d\vec{x} \\
&= (\sigma_f^2 Z(\lambda))^2\ \mathcal{N}(\vec{z}'|\vec{z}, 2\lambda)\ \int \mathcal{N}(\vec{x}|\frac{1}{2}(\vec{z} + \vec{z}'), \frac{\lambda}{2})\mathcal{N}(\vec{x}|\vec{\mu}, S)d\vec{x} \\
&= (\sigma_f^2 Z(\lambda))^2\ \mathcal{N}(\vec{z}'|\vec{z}, 2\lambda)\mathcal{N}(\vec{\mu}|\frac{1}{2}(\vec{z} + \vec{z}'), \frac{\lambda}{2} + S) \\
&= \sigma_f^4(2\pi)^Q(\prod_{q=1}^{Q}\lambda_q)\ \mathcal{N}(\vec{z}'|\vec{z}, 2\lambda)\ \mathcal{N}(\vec{\mu}|\frac{\vec{z} + \vec{z}'}{2}, \frac{\lambda}{2} + S)
\end{aligned} \tag{A17}$$

For the case of a diagonal covariance matrix S the Ψ_2 statistic can be simplified further [16].

Appendix C. vCGPDM Dynamics ELBO Derivation

We now present a detailed derivation of the ELBO with a focus on the dynamics component. Assume we deal with M parts. We have $M \times M$ latent dynamics mappings, which are combined into M mappings with product of experts—multiplying and renormalizing the distributions from all parts' predictions about each part. Each of the $M \times M$ mappings $f^{j,i}()$ is augmented with IPs $\vec{z}_{:}^{j,i}$ and IVs $\vec{u}_{:}^{j,i}$, which are drawn out of the same $\mathcal{GP}$ priors as the mappings. For clarity, we omit spelling out the dependence of the IVs on the IPs in the following, and we ask the reader to remember that any distribution over IVs is implicitly conditioned onto the corresponding IPs. The full augmented joint distribution of the model, which is derived in Section 4 (Equation (5)), is:

$$\begin{aligned}
p(\vec{y}_{:}, \vec{g}_{:}, \vec{x}_{:}, \vec{v}_{:}, \vec{f}_{:}^{:,:}, \vec{u}_{:}^{:,:}) &= p(\vec{y}_{:}|\vec{g}_{:})p(\vec{g}_{:}|\vec{x}_{:}, \vec{v}_{:})p(\vec{v}_{:})p(\vec{x}_{:}, \vec{f}_{:}^{:,:}|\vec{u}_{:}^{:,:})p(\vec{u}_{:}^{:,:}) \\
&= \left[\prod_{i=1}^{M} p(\vec{y}_{:}^{i}|\vec{g}_{:}^{i})p(\vec{g}_{:}^{i}|\vec{x}_{:}^{i}, \vec{v}_{:}^{i})p(\vec{v}_{:}^{i})\right] p(\vec{x}_{:}, \vec{f}_{:}^{:,:}|\vec{u}_{:}^{:,:})p(\vec{u}_{:}^{:,:}) \\
&= \left[\prod_{i=1}^{M}\left[\prod_{d=1}^{D_i} p((\vec{y}_{:}^{i})_d|(\vec{g}_{:}^{i})_d)p((\vec{g}_{:}^{i})_d|\vec{x}_{:}^{i}, (\vec{v}_{:}^{i})_d)p((\vec{v}_{:}^{i})_d)\right]\right] \\
&\times \left[\prod_{t=1}^{T}\left[\prod_{i=1}^{M} p(\vec{x}_t^{i}|\vec{f}_t^{:,i}, \alpha^{:,i})\right]\left[\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{f}_t^{j,i}|\vec{f}_{1:t-1}^{j,i}, \vec{x}_{0:t-1}^{j}, \vec{u}_{:}^{j,i})\right]\right] \\
&\times \left[\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{u}_{:}^{j,i})\right]\left[\prod_{i=1}^{M} p(\vec{x}_0^{i})\right]
\end{aligned} \tag{A18}$$

The full proposal variational posterior is (Equation (12) in Section 4):

$$\begin{aligned} q(\vec{g}_{:}^{:},\vec{x}_{:}^{:},\vec{v}_{:}^{:},\vec{f}_{:}^{:,:},\vec{u}_{:}^{:,:}) &= p(\vec{g}_{:}^{:}|\vec{x}_{:}^{:},\vec{v}_{:}^{:})q(\vec{v}_{:}^{:})p(\vec{f}_{:}^{:,:}|\vec{x}_{:}^{:},\vec{u}_{:}^{:,:})q(\vec{x}_{:}^{:})q(\vec{u}_{:}^{:,:}) \\ &= p(\vec{g}_{:}^{:}|\vec{x}_{:}^{:},\vec{v}_{:}^{:})q(\vec{v}_{:}^{:}) \left[\prod_{t=1}^{T}\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{f}_{t}^{j,i}|\vec{f}_{1:t-1}^{j,i},\vec{x}_{0:t-1}^{j},\vec{u}_{:}^{j,i})\right] q(\vec{x}_{:}^{:})q(\vec{u}_{:}^{:,:}) \end{aligned} \tag{A19}$$

Thus, the ELBO is given by:

$$\mathcal{L}(\Theta) = \int d\vec{g}_{:}^{:}\, d\vec{x}_{:}^{:}\, d\vec{v}_{:}^{:}\, d\vec{f}_{:}^{:,:}\, d\vec{u}_{:}^{:,:} q(\vec{g}_{:}^{:},\vec{x}_{:}^{:},\vec{v}_{:}^{:},\vec{f}_{:}^{:,:},\vec{u}_{:}^{:,:}) \log\left(\frac{p(\vec{y}_{:}^{:},\vec{g}_{:}^{:},\vec{x}_{:}^{:},\vec{v}_{:}^{:},\vec{f}_{:}^{:,:},\vec{u}_{:}^{:,:})}{q(\vec{g}_{:}^{:},\vec{x}_{:}^{:},\vec{v}_{:}^{:},\vec{f}_{:}^{:,:},\vec{u}_{:}^{:,:})}\right) = \sum_{i=1}^{M}\mathcal{L}_{kin}^{i} + \mathcal{L}_{dyn} \tag{A20}$$

$$= \sum_{i=1}^{M}\sum_{d=1}^{D}\int d\vec{x}_{:}^{i} d\vec{v}_{:}^{i} d(\vec{g}_{:}^{i})_d p((\vec{g}_{:}^{i})_d|\vec{x}_{:}^{i},\vec{v}_{:}^{i})q(\vec{x}_{:}^{i})q(\vec{v}_{:}^{i}) \log\frac{p((\vec{y}_{:}^{i})_d|(\vec{g}_{:}^{i})_d)}{q(\vec{v}_{:}^{i})} \tag{A21}$$

$$+ \int d\vec{u}_{:}^{:,:} q(\vec{u}_{:}^{:,:}) \left[\sum_{t=1}^{T}\int q(\vec{x}_{t}^{:})q(\vec{x}_{-t}^{:}) \left(\int d\vec{f}_{t}^{:,:} \left[\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{f}_{t}^{j,i}|\vec{f}_{1:t-1}^{j,i},\vec{x}_{0:t-1}^{j},\vec{u}_{:}^{j,i})\right] \log\prod_{i=1}^{M} p(\vec{x}_{t}^{i}|\vec{f}_{t}^{:,i},\alpha^{:,i})\right)\right]$$

$$+ \int d\vec{u}_{:}^{:,:} q(\vec{u}_{:}^{:,:}) \log\frac{p(\vec{u}_{:}^{:,:})}{q(\vec{u}_{:}^{:,:})} + \sum_{t=0}^{1}\int q(\vec{x}_{t}^{:}) \log p(\vec{x}_{t}^{:}) d\vec{x}_{t}^{:} + H(q(\vec{x}_{:}^{:})) \tag{A22}$$

The term in Equation (A21), which we call $\sum_{i=1}^{M}\mathcal{L}_{kin}^{i}$, is the GPLVM ELBO up to $H(q(\vec{x}_{:}^{:}))$ and is given in [16]. Next, we consider only the ELBO component which is relevant for the dynamics $\mathcal{L}_{dyn}$ (last two lines of the right hand side of Equation (A22)) and apply the sufficient statistics assumption: knowing $\vec{x}_{-t}^{j}$ and $\vec{u}_{:}^{j,i}$ is sufficient for the $\vec{f}_{t}^{j,i}$ distribution, i.e., $p(\vec{f}_{t}^{j,i}|\vec{f}_{1:t-1}^{j,i},\vec{x}_{0:t-1}^{j},\vec{u}_{:}^{j,i}) = p(\vec{f}_{t}^{j,i}|\vec{x}_{0:t-1}^{j},\vec{u}_{:}^{j,i})$. This assumption lower-bounds $\mathcal{L}_{dyn}$, because it constrains the variational posterior in Equation (A19) away from the correct solution. The sum over the initial latent points in the last line of Equation (A22) may be longer or shorter depending on the dynamics model order, here we use a second order model. The innermost integral can then be written as:

$$\begin{aligned} \mathcal{A} &= \int \left[\prod_{i=1}^{M}\prod_{j=1}^{M} p(\vec{f}_{t}^{j,i}|\vec{x}_{-t}^{j},\vec{u}_{:}^{j,i})\right] \log\prod_{i=1}^{M} p(\vec{x}_{t}^{i}|\vec{f}_{t}^{:,i},\alpha^{:,i}) d\vec{f}_{t}^{:,:} \\ &= \sum_{i=1}^{M}\int \left[\prod_{j=1}^{M} p(\vec{f}_{t}^{j,i}|\vec{x}_{-t}^{j},\vec{u}_{:}^{j,i})\right] \log p(\vec{x}_{t}^{i}|\vec{f}_{t}^{:,i},\alpha^{:,i}) d\vec{f}_{t}^{:,i} \\ &= \sum_{i=1}^{M}\int \left[\prod_{j=1}^{M} \mathcal{N}(\vec{f}_{t}^{j,i}|\vec{\mu}_{\vec{f}_{t}^{j,i}},\mathbf{S}_{\vec{f}_{t}^{j,i}})\right] \log\mathcal{N}(\vec{x}_{t}^{i}|\alpha^{i}\sum_{j=1}^{M}(\alpha^{j,i})^{-1}\vec{f}_{t}^{j,i},\mathbf{I}\alpha^{i}) d\vec{f}_{t}^{:,i} \\ &= \sum_{i=1}^{M}\left[-\tfrac{1}{2}\,\mathrm{tr}\left[\alpha^{i}\sum_{j=1}^{M}(\alpha^{j,i})^{-2}\mathbf{S}_{\vec{f}_{t}^{j,i}}\right] + \log\mathcal{N}\left(\vec{x}_{t}^{i}|\alpha^{i}\sum_{j=1}^{M}(\alpha^{j,i})^{-1}\vec{\mu}_{\vec{f}_{t}^{j,i}},\mathbf{I}\alpha^{i}\right)\right] \end{aligned} \tag{A23}$$

$$\vec{\mu}_{\vec{f}_{t}^{j,i}} = \mathbf{K}_{\vec{x}_{-t}^{j},\vec{z}_{:}^{j,i}}^{j,i}\left(\mathbf{K}_{\vec{z}_{:}^{j,i},\vec{z}_{:}^{j,i}}^{j,i}\right)^{-1}\vec{u}_{:}^{j,i} \tag{A24}$$

$$\mathbf{S}_{\vec{f}_{t}^{j,i}} = \mathbf{K}_{\vec{x}_{-t}^{j},\vec{x}_{-t}^{j}}^{j,i} - \mathbf{K}_{\vec{x}_{-t}^{j},\vec{z}_{:}^{j,i}}^{j,i}\left(\mathbf{K}_{\vec{z}_{:}^{j,i},\vec{z}_{:}^{j,i}}^{j,i}\right)^{-1}\mathbf{K}_{\vec{x}_{-t}^{j},\vec{z}_{:}^{j,i}}^{j,i} \tag{A25}$$

$$\alpha^{i} = \left(\sum_{j=1}^{M}(\alpha^{j,i})^{-1}\right)^{-1} \tag{A26}$$

where $\mathbf{K}$ are kernel matrices, obtained by evaluating the kernel function at the pairs of points indicated by the subscripts. Equations (A24) and (A25) follow from the standard formulas for conditional Gaussians (see, e.g., [45], Identities 352–254). Equation (A26) follows from the product-of-experts construction (see Section 3, Equation (1)).

The next step in the evaluation of Equation (A22) is integrating $\mathcal{A}$ over $d\vec{x}^{:}_{t}$ and $\vec{x}^{:}_{-t}$, which requires averaging kernel matrices. The results of this averaging are denoted by Ψ (see Appendix B and [16] for a derivation). We denote $\Psi^{j,i}_{0}(\vec{x}^{j}_{-t}) = \int d\vec{x}^{j}_{-t} q(\vec{x}^{j}_{-t}) \mathbf{K}^{j,i}_{\vec{x}^{j}_{-t},\vec{x}^{j}_{-t}}$, etc.:

$$
\begin{aligned}
\mathcal{B} &= \int q(\vec{x}^{:}_{t}) q(\vec{x}^{:}_{-t})\ \mathcal{A}\ d\vec{x}^{:}_{t}\ d\vec{x}^{:}_{-t} \\
&= \sum_{i=1}^{M} \Bigg(-\frac{1}{2}\alpha_{i} \sum_{j=1}^{M} (\alpha^{j,i})^{-2} \operatorname{tr} \left[\Psi^{j,i}_{0}(\vec{x}^{j}_{-t}) - \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \Psi^{j,i}_{2}(\vec{x}^{j}_{-t}) \right] \\
&- Q_{i} \log \sqrt{2\pi\alpha^{i}} - \frac{1}{2}(\alpha^{i})^{-1} \left[\operatorname{tr}(\mathbf{S}_{\vec{x}^{i}_{t}}) + \vec{\mu}^{T}_{\vec{x}^{i}_{t}} \vec{\mu}_{\vec{x}^{i}_{t}} \right] \\
&+ \vec{\mu}^{T}_{\vec{x}^{i}_{t}} \Big(\sum_{j=1}^{M} (\alpha^{j,i})^{-1} \Psi^{j,i}_{1}(\vec{x}^{j}_{-t}) \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \vec{u}^{j,i}_{:} \Big) \\
&- \frac{1}{2}\alpha^{i} \sum_{j=1}^{M} \sum_{k=1}^{M} (\alpha^{j,i})^{-1} (\alpha^{k,i})^{-1} \vec{u}^{j,i\,T}_{:} \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \Psi^{j,k,i}_{2}(\vec{x}^{j}_{-t}, \vec{x}^{k}_{-t}) \left(\mathbf{K}^{k,i}_{\vec{z}^{k,i}_{:},\vec{z}^{k,i}_{:}} \right)^{-1} \vec{u}^{k,i}_{:} \Bigg)
\end{aligned}
\tag{A27}
$$

Next, we sum this expression over time points:

$$
\begin{aligned}
\mathcal{C} &= \sum_{t=1}^{T} \mathcal{B} \\
&= \sum_{i=1}^{M} \Bigg(\sum_{t=1}^{T} \Bigg[-\frac{1}{2}\alpha^{i} \sum_{j=1}^{M} (\alpha^{j,i})^{-2} \operatorname{tr} \left[\Psi^{j,i}_{0}(\vec{x}^{j}_{-t}) - \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \Psi^{j,i}_{2}(\vec{x}^{j}_{-t}) \right] \\
&- \log \sqrt{2\pi\alpha^{i}} - \frac{1}{2}(\alpha^{i})^{-1} \left[\operatorname{tr}(\mathbf{S}_{\vec{x}^{i}_{t}}) + \vec{\mu}^{T}_{\vec{x}^{i}_{t}} \vec{\mu}_{\vec{x}^{i}_{t}} \right] \Bigg] \\
&+ \sum_{j=1}^{M} (\alpha^{j,i})^{-1} \left[\sum_{t=1}^{T} \vec{\mu}^{T}_{\vec{x}^{i}_{t}} \Psi^{j,i}_{1}(\vec{x}^{j}_{-t}) \right] \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \vec{u}^{j,i}_{:} \\
&- \frac{1}{2}\alpha^{i} \sum_{j=1}^{M} \sum_{k=1}^{M} (\alpha^{j,i})^{-1} (\alpha^{k,i})^{-1} \vec{u}^{j,i\,T}_{:} \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \left[\sum_{t=1}^{T} \Psi^{j,k,i}_{2}(\vec{x}^{j}_{-t}, \vec{x}^{k}_{-t}) \right] \left(\mathbf{K}^{j,i}_{\vec{z}^{k,i}_{:},\vec{z}^{k,i}_{:}} \right)^{-1} \vec{u}^{k,i}_{:} \Bigg)
\end{aligned}
\tag{A28}
$$

For every part $i \in 1 \ldots M$, we stack up the $\vec{u}^{:,i}_{:}$ into $\vec{u}^{i}$ (first by IV-index $k = 1, \ldots, K$, and then by part index such that $\vec{u}^{j,i}_{k} = (\vec{u}^{i})_{Q^{i} \cdot (K \cdot j + k) + q^{i} + 1}$) and construct a large block matrices $\mathcal{F}^{i}$ and stacked vector $\mathcal{G}^{i}$ with block elements

$$
\mathcal{F}^{i}_{j,k} = \alpha^{i} \alpha^{-1}_{j,i} \alpha^{-1}_{k,i} \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \left[\sum_{t=1}^{T} \Psi^{j,k,i}_{2}(\vec{x}^{j}_{-t}, \vec{x}^{k}_{-t}) \right] \left(\mathbf{K}^{k,i}_{\vec{z}^{k,i}_{:},\vec{z}^{k,i}_{:}} \right)^{-1}
\tag{A29}
$$

$$
\mathcal{G}^{i}_{j} = \left(\alpha^{-1}_{j,i} \left[\sum_{t=1}^{T} \vec{\mu}^{T}_{\vec{x}^{i}_{t}} \Psi^{j,i}_{1}(\vec{x}^{j}_{-t}) \right] \left(\mathbf{K}^{j,i}_{\vec{z}^{j,i}_{:},\vec{z}^{j,i}_{:}} \right)^{-1} \right)^{T}
\tag{A30}
$$

For $j \neq k$: $\Psi^{j,k,i}_{2}(\vec{x}^{j}_{-t}, \vec{x}^{k}_{-t}) = \Psi^{j,i}_{1}(\vec{x}^{j}_{-t}) \Psi^{k,i}_{1}(\vec{x}^{k}_{-t})$. Otherwise, $\Psi^{j,j,i}_{2}(\vec{x}^{j}_{-t}, \vec{x}^{j}_{-t}) = \Psi^{j,i}_{2}(\vec{x}^{j}_{-t})$. We rewrite $\mathcal{C}$ as a quadratic form in the stacked augmenting IVs $\vec{u}^{i}$ to facilitate closed-form optimization of the dynamics ELB in Equation (A22) with respect to the stacked IV density $q(\vec{u}^{:})$ using variational calculus, as described in Appendix A:

$$\mathcal{C} = \sum_{i=1}^{M} \left[-\frac{1}{2} \vec{u}^{iT} \mathcal{F}^i \vec{u}^i + \vec{u}^{iT} \mathcal{G}^i + \mathcal{H}^i \right]$$

$$= \sum_{i=1}^{M} \left[-\frac{1}{2} (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)^T \mathcal{F}^i (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i) + \frac{1}{2} \mathcal{G}^{iT} \mathcal{F}^{i-1} \mathcal{G}^i + \mathcal{H}^i \right] \tag{A31}$$

$$\mathcal{C} = \sum_{i=1}^{M} \mathcal{C}^i \tag{A32}$$

$$\mathcal{C}^i = -\frac{1}{2} (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)^T \mathcal{F}^i (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i) + \frac{1}{2} \mathcal{G}^{iT} \mathcal{F}^{i-1} \mathcal{G}^i + \mathcal{H}^i \tag{A33}$$

$$\mathcal{H}^i = \sum_{t=1}^{T} \left[-\frac{1}{2} \alpha_i \sum_{j=1}^{M} (\alpha_{j,i}^{-1})^2 \operatorname{tr} \left[\Psi_0^{j,i}(\vec{x}^j_{-t}) - \left(\mathbf{K}^{j,i}_{\vec{z}_{:}^{j,i},\vec{z}_{:}^{j,i}} \right)^{-1} \Psi_2^{j,i}(\vec{x}^j_{-t}) \right] \right.$$
$$\left. - \log \sqrt{2\pi\alpha^i} - \frac{1}{2} \alpha_i^{-1} \left[\operatorname{tr}(\mathbf{S}_{\vec{x}_t^i}) + \vec{\mu}_{\vec{x}_t^i}^T \vec{\mu}_{\vec{x}_t^i} \right] \right] \tag{A34}$$

After this optimization, we can write the dynamics ELBO using $p(\vec{u}^i) = \prod_{j=1}^{M} p(\vec{u}_{:}^{j,i}) = \prod_{j=1}^{M} \mathcal{N}(\vec{u}^{j,i}|0, \mathbf{K}_{\vec{z}_{:}^{j,i},\vec{z}_{:}^{j,i}}) = \mathcal{N}(\vec{u}^i|0, \mathbf{K}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}})$ where $\mathbf{K}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}}$ is a block-diagonal covariance matrix with the blocks given by the individual $\mathbf{K}_{\vec{z}_{:}^{j,i},\vec{z}_{:}^{j,i}}$:

$$\mathcal{L}_{dyn}(\Theta) \geq \log \int p(\vec{u}_{:}^{:,:}) \exp(\mathcal{C}) d\vec{u}_{:}^{:,:} + H(q(\vec{x}_{:}^{:}))$$

$$= \log \prod_{i=1}^{M} \int p(\vec{u}^i) \exp(\mathcal{C}^i) d\vec{u}^i + H(q(\vec{x}_{:}^{:}))$$

$$= \sum_{i=1}^{M} \left[\log \int p(\vec{u}^i) \exp(-\frac{1}{2} (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)^T \mathcal{F}^i (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)) d\vec{u}^i + \frac{1}{2} \mathcal{G}^{iT} \mathcal{F}^{i-1} \mathcal{G}^i + \mathcal{H}^i \right] + H(q(\vec{x}_{:}^{:})) \tag{A35}$$

$$= \sum_{i=1}^{M} \left[-\log \sqrt{(2\pi)^{\dim(\mathcal{F}^{i-1})} |\mathcal{F}^{i-1} + \mathbf{K}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}}|} - \frac{1}{2} \mathcal{G}^{iT} \mathcal{F}^{i-1} (\mathcal{F}^{i-1} + \mathbf{K}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}})^{-1} \mathcal{F}^{i-1} \mathcal{G}^i + \log Z(\mathcal{F}^{i-1}) \right]$$

$$+ \sum_{i=1}^{M} \left[\frac{1}{2} \mathcal{G}^{iT} \mathcal{F}^{i-1} \mathcal{G}^i + \mathcal{H}^i \right] + H(q(\vec{x}_{:}^{:}))$$

This is the expression which we optimize with respect to $q(\vec{x}_{:}^{:})$ and $q(\vec{u}_{:}^{:,:})$. Since the stacked dynamics IVs $\vec{u}^i$ do not interact across parts in this expression (Line 2), it follows that density $q(\vec{u}_{:}^{:,:})$ factorizes across parts. Their optimal density for each part is given by (cf. Equation (A6), Z is the normalization constant of the multivariate Gaussian):

$$q(\vec{u}^i) = \frac{1}{Z} p(\vec{u}^i) \exp(\mathcal{C}^i)$$

$$= \frac{1}{Z} \mathcal{N}(\vec{u}^i|0, \mathbf{K}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}}) \exp(-\frac{1}{2} (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)^T \mathcal{F}^i (\vec{u}^i - \mathcal{F}^{i-1} \mathcal{G}^i)) \tag{A36}$$

$$= \mathcal{N}(\vec{u}^i|(\mathbf{K}^{-1}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}} + \mathcal{F}^i)^{-1} \mathcal{G}^i, (\mathbf{K}^{-1}_{\vec{z}_{:}^{:,i},\vec{z}_{:}^{:,i}} + \mathcal{F}^i)^{-1})$$

References

1. Bizzi, E.; Cheung, V.; d'Avella, A.; Saltiel, P.; Tresch, M. Combining modules for movement. *Brain Res. Rev.* **2008**, *57*, 125–133. [CrossRef] [PubMed]
2. Endres, D.; Chiovetto, E.; Giese, M. Model selection for the extraction of movement primitives. *Front. Comput. Neurosci.* **2013**, *7*, 185. [CrossRef] [PubMed]
3. Schaal, S. Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics. In *Adaptive Motion of Animals and Machines*; Kimura, H., Tsuchiya, K., Ishiguro, A., Witte, H., Eds.; Springer: Tokyo, Japan, 2006; pp. 261–280. [CrossRef]

4. Giese, M.A.; Mukovskiy, A.; Park, A.N.; Omlor, L.; Slotine, J.J.E. Real-Time Synthesis of Body Movements Based on Learned Primitives. In *Statistical and Geometrical Approaches to Visual Motion Analysis*; Cremers, D., Rosenhahn, B., Yuille, A.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5604, pp. 107–127.
5. Koch, K.H.; Clever, D.; Mombaur, K.; Endres, D.M. Learning Movement Primitives from Optimal and Dynamically Feasible Trajectories for Humanoid Walking. In Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids 2015), Seoul, Korea, 3–5 November 2015; pp. 866–873.
6. Chiovetto, E.; Berret, B.; Pozzo, T. Tri-dimensional and triphasic muscle organization of whole-body pointing movements. *Neuroscience* **2010**, *170*, 1223–1238. [CrossRef] [PubMed]
7. Omlor, L.; Giese, M.A. Anechoic Blind Source Separation using Wigner Marginals. *J. Mach. Learn. Res.* **2011**, *12*, 1111–1148.
8. Chiovetto, E.; Giese, M.A. Kinematics of the coordination of pointing during locomotion. *PLoS ONE* **2013**, *8*, e79555. [CrossRef] [PubMed]
9. Clever, D.; Harant, M.; Koch, K.H.; Mombaur, K.; Endres, D.M. A novel approach for the generation of complex humanoid walking sequences based on a combination of optimal control and learning of movement primitives. *Robot. Autom. Syst.* **2016**, *83*, 287–298. [CrossRef]
10. Mussa-Ivaldi, F.A.; Solla, S.A. Neural Primitives for Motion Control. *IEEE J. Ocean. Eng.* **2004**, *29*, 640–650. [CrossRef]
11. Hart, C.B.; Giszter, S. Distinguishing Synchronous and Time Varying Synergies using Point Process Interval Statistics: Motor Primitives in Frog and Rat. *Front. Comput. Neurosci.* **2013**, *7*, 52. [CrossRef] [PubMed]
12. Ijspeert, A.J.; Nakanishi, J.; Hoffmann, H.; Pastor, P.; Schaal, S. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Comput.* **2013**, *25*, 328–373. [CrossRef] [PubMed]
13. Rückert, E.; d'Avella, A. Learned Parameterized Dynamic Movement Primitives with Shared Synergies for Controlling Robotic and Musculoskeletal Systems. *Front. Comput. Neurosci.* **2013**, *7*, 138. [CrossRef] [PubMed]
14. Velychko, D.; Endres, D.; Taubert, N.; Giese, M.A. Coupling Gaussian Process Dynamical Models with Product-of-Experts Kernels. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2014; pp. 603–610. [CrossRef]
15. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*; The MIT Press: Cambridge, MA, USA, 2005.
16. Titsias, M.K.; Lawrence, N.D. Bayesian Gaussian Process Latent Variable Model. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy, 13–15 May 2010; pp. 844–851.
17. Bizzi, E.; Cheung, V.C. The Neural Origin of Muscle Synergies. *Front. Comput. Neurosci.* **2013**, *7*, 51. [CrossRef] [PubMed]
18. Földiák, P.; Endres, D. Sparse coding. *Scholarpedia* **2008**, *3*, 2984. [CrossRef]
19. Velychko, D.; Knopp, B.; Endres, D. The variational coupled Gaussian Process Dynamical Model. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2017; pp. 291–299. [CrossRef]
20. Candela, J.Q.; Rasmussen, C.E. A Unifying View of Sparse Approximate Gaussian Process Regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.
21. Snelson, E.; Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*; MIT Press: Cambridge, MA, USA, 2006; pp. 1257–1264.
22. Lawrence, N.D. Learning for Larger Datasets with the Gaussian Process Latent Variable Model. *Artif. Intell. Stat.* **2007**, *2*, 243–250.
23. Titsias, M.K. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *Artif. Intell. Stat.* **2009**, *5*, 567–574.
24. Damianou, A.C.; Titsias, M.; Lawrence, N.D. Variational Gaussian Process Dynamical Systems. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K., Eds.; MIT Press: Cambridge, MA, USA, 2011; pp. 2510–2518.
25. Wang, J.M.; Fleet, D.J.; Hertzmann, A. Gaussian Process Dynamical Models for Human Motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 283–298. [CrossRef] [PubMed]
26. Urtasun, R.; Fleet, D.J.; Lawrence, N.D. Modeling Human Locomotion with Topologically Constrained Latent Variable Models. In *Human Motion–Understanding, Modeling, Capture and Animation*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4814, pp. 104–118.

27. Taubert, N.; Endres, D.; Christensen, A.; Giese, M.A. Shaking Hands in Latent Space. In *Annual Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7006, pp. 330–334.
28. Levine, S.; Wang, J.M.; Haraux, A.; Popović, Z.; Koltun, V. Continuous Character Control with Low-Dimensional Embeddings. *ACM Trans. Graph.* **2012**, *31*, 28. [CrossRef]
29. Chen, J.; Kim, M.; Wang, Y.; Ji, Q. Switching Gaussian Process Dynamic Models for simultaneous composite motion tracking and recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2009 (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 2655–2662. [CrossRef]
30. Mattos, C.L.C.; Dai, Z.; Damianou, A.; Forth, J.; Barreto, G.A.; Lawrence, N.D. Recurrent Gaussian Processes. *arXiv* **2015**, arxiv:1511.06644.
31. Frigola, R.; Lindsten, F.; Schön, T.B.; Rasmussen, C. Bayesian Inference and Learning in Gaussian Process State-Space Models with Particle MCMC. In *Advances in Neural Information Processing Systems 26*; Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 3156–3164.
32. Frigola, R.; Chen, Y.; Rasmussen, C. Variational Gaussian Process State-Space Models. In *Advances in NIPS 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K., Eds.; NIPS Foundation: Montreal, QC, Canada, 2014; pp. 3680–3688.
33. Bauer, M.; van der Wilk, M.; Rasmussen, C. Understanding Probabilistic Sparse Gaussian Process Approximations. *arXiv* **2016**, arxiv:1606.04820.
34. Taubert, N.; Christensen, A.; Endres, D.; Giese, M. Online simulation of emotional interactive behaviors with hierarchical Gaussian process dynamical models. In Proceedings of the ACM Symposium on Applied Perception (SAP '12), Los Angeles, CA, USA, 3–4 August 2012; pp. 25–32. [CrossRef]
35. Hinton, G.E. Products of Experts. In Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99), Edinburgh, UK, 7–10 September 1999; Volume 1, pp. 1–6.
36. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: Secaucus, NJ, USA, 2006.
37. Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I.J.; Bergeron, A.; Bouchard, N.; Bengio, Y. Theano: New features and speed improvements. *arXiv* **2012**, arxiv:1211.5590.
38. Jones, E.; Oliphant, T.; Peterson, P. SciPy: Open Source Scientific Tools for Python. Available online: https://www.scipy.org/ (accessed on 9 October 2015).
39. Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [CrossRef]
40. Taubert, N.; Löffler, M.; Ludolph, N.; Christensen, A.; Endres, D.; Giese, M. A virtual reality setup for controllable, stylized real-time interactions between humans and avatars with sparse Gaussian process dynamical models. In Proceedings of the ACM Symposium on Applied Perception (SAP '13), Dublin, Ireland, 22–23 August 2013; pp. 41–44. [CrossRef]
41. Paraschos, A.; Daniel, C.; Peters, J.; Neumann, G. Probabilistic Movement Primitives. In *Advances in NIPS 26*; Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K., Eds.; NIPS Foundation: Montreal, QC, Canada, 2013; pp. 2616–2624.
42. Pastor, P.; Kalakrishnan, M.; Righetti, L.; Schaal, S. Towards Associative Skill Memories. In Proceedings of the 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka, Japan, 29 November–1 December 2012; pp. 309–315.
43. Land, W.M.; Rosenbaum, D.A.; Seegelke, C.; Schack, T. Whole-body posture planning in anticipation of a manual prehension task: Prospective and retrospective effects. *Acta Psychol.* **2013**, *144*, 298–307. [CrossRef] [PubMed]
44. Deisenroth, M.; Fox, D.; Rasmussen, C. Gaussian Processes for Data-Efficient Learning in Robotics and Control. *Pattern Anal. Mach. Intell. IEEE Trans.* **2015**, *37*, 408–423. [CrossRef] [PubMed]
45. Petersen, K.B.; Pedersen, M.S. *The Matrix Cookbook*; Version 20121115; Technical University of Denmark: Lyngby, Denmark, 2012.